

The XMM-Newton ABC Guide:

An Introduction to XMM-Newton Data Analysis

NASA/GSFC XMM-Newton Guest Observer Facility

Steve Snowden, Lynne Valencic, Brendan Perry, Michael Arida, K.D.
Kuntz

With contributions by: Ilana Harrus, Stefan Immmler, Rick Shafer,
Randall Smith, Martin Still

Version 4.5
for XMM-SAS v13.0

April 2014

Copies of this guide are available in `html`, `postscript` and `pdf` formats.

Contents

1	Introduction	1
1.1	ACKNOWLEDGMENTS	1
2	Useful Information and References	2
2.1	MAIN WEB SITES	2
2.2	<i>XMM-NEWTON</i> HELP DESKS	2
2.3	MISSION PLANNING AND SPACECRAFT STATUS	2
2.4	PUBLIC DATA ARCHIVES	3
2.5	CALIBRATION DATA	3
2.6	SOFTWARE	3
2.7	ANALYSIS, DOCUMENTATION AND HELPFUL HINTS	3
3	Data	4
3.1	USEFUL DOCUMENTATION	4
3.2	THE DATA	4
3.3	PI Data	4
3.3.1	ODF Data	5
3.3.2	Pipeline Product Data – Summary Files and Groupings	5
3.3.3	Pipeline Product Data – Data Files	6
4	Setting Up and Running SAS	10
4.1	Installation	10
4.2	Calibration Data	10
4.3	SAS Environmental Parameters and Invocation	10
4.3.1	SAS Helpful Hints	11
4.4	SAS Syntax and Logic	11
4.4.1	Command Line Syntax	11
4.4.2	Table Syntax	12
4.4.3	Filtering Logic	12
5	Preparing the Data for Processing	13
5.1	cifbuild	13
5.2	odfingest	14
6	An EPIC Data Processing and Analysis Primer	15
6.1	Rerun the Pipeline on the Command Line	16
6.2	Examine and Analyze Data on the Command Line (Imaging Mode)	17
6.2.1	Create and Display an Image	17
6.2.2	Create and Display a Light Curve	19
6.2.3	Applying Standard Filters the Data	19
6.2.4	Applying Time Filters the Data	21
6.2.5	Source Detection with <i>edetect_chain</i>	22
6.2.6	Extract the Source and Background Spectra	23
6.2.7	Determine the Spectrum Extraction Areas	25
6.2.8	Check for Pile Up	25
6.2.9	My Observation is Piled Up! Now What?	26

6.2.10	Create the Photon Redistribution Matrix (RMF) and Ancillary File (ARF)	29
6.3	Examine and Analyze Data on the Command Line (Timing Mode)	30
6.3.1	Create and Display an Image	30
6.3.2	Create and Display a Light Curve	31
6.3.3	Applying Standard Filters to the Data	31
6.3.4	Applying Time Filters to the Data	32
6.3.5	Extract the Source and Background Spectra	32
6.3.6	Determine the Spectrum Extraction Areas	33
6.3.7	Check for Pile Up	33
6.3.8	My Timing Mode Data is Piled Up! Now What?	33
6.3.9	Create the Photon Redistribution Matrix (RMF) and Ancillary File (ARF)	34
6.4	Rerun the Pipeline with the GUI	37
6.5	Examine and Analyze Data with the GUI (Imaging Mode)	37
6.5.1	An Introduction to <i>xmmselect</i>	37
6.5.2	Create and Display an Image	38
6.5.3	Create and Display a Light Curve	38
6.5.4	Applying Standard Filters the Data	40
6.5.5	Applying Time Filters the Data	41
6.5.6	Source Detection with <i>edetect_chain</i>	42
6.5.7	Extract the Source and Background Spectra	43
6.5.8	Determine the Spectrum Extraction Areas	44
6.5.9	Check for Pile Up	45
6.5.10	My Observation is Piled Up! Now What?	48
6.5.11	Create the Photon Redistribution Matrix (RMF) and Ancillary File (ARF)	48
6.6	Examine and Analyze Data with the GUI (Timing Mode)	49
6.6.1	Create and Display an Image	49
6.6.2	Create and Display a Light Curve	50
6.6.3	Applying Standard Filters to the Data	50
6.6.4	Applying Time Filters to the Data	51
6.6.5	Extract the Source and Background Spectra	51
6.6.6	Determine the Spectrum Extraction Areas	52
6.6.7	Check for Pile Up	52
6.6.8	My Timing Mode Data is Piled Up! Now What?	53
6.6.9	Create the Photon Redistribution Matrix (RMF) and Ancillary File (ARF)	55
6.7	Prepare and Fit the Spectrum	55
6.8	Timing Analysis	56
6.9	Once More, This Time with Feeling and Ftools	60
7	An RGS Data Processing and Analysis Primer	62
7.1	Rerun the Pipeline on the Command Line	63
7.1.1	Potentially useful tips for using the pipeline	63
7.1.2	A Nearby Bright Optical Source	63
7.1.3	A Nearby Bright X-ray Source	64
7.1.4	User-defined Source Coordinates	64
7.2	Examine and Analyze the Data on the Command Line	64
7.2.1	Create and Display an Image	64
7.2.2	Create and Display a Light Curve	65
7.2.3	Generating the Good Time Interval (GTI) File	66
7.2.4	Applying the GTI	67
7.2.5	Creating the Response Matrices (RMFs)	67
7.2.6	Combining Spectra	68
7.3	Rerun the Pipeline with the GUI	68
7.3.1	Potentially useful tips for using the pipeline	69
7.3.2	A Nearby Bright Optical Source	69
7.3.3	A Nearby Bright X-ray Source	69
7.3.4	User-defined Source Coordinates	69
7.4	Examine and Analyze the Data with the GUI	70

7.4.1	An Introduction to <i>xmmselect</i>	70
7.4.2	Create and Display an Image	71
7.4.3	Create and Display a Light Curve	71
7.4.4	Generating the Good Time Interval (GTI) File	72
7.4.5	Applying the GTI	73
7.4.6	Creating the Response Matrices (RMFs)	73
7.4.7	Combining Spectra	74
7.5	Approaches to Spectral Fitting and the Cash Statistic	74
7.5.1	Spectral Rebinning	75
7.6	Fitting a Spectral Model	76
7.7	Analysis of Extended Sources	76
7.7.1	Region masks	76
7.7.2	Fitting spectral models to extended sources	78
7.7.3	Model limitations	79
8	An OM Data Processing and Analysis Primer	80
8.1	OM Artifacts and General Information	80
8.2	Imaging Mode	82
8.2.1	Rerunning the Pipeline	82
8.2.2	Verifying the Output	82
8.3	Fast Mode	84
8.3.1	Rerunning the Pipeline	84
8.3.2	Verifying the Output	85
8.4	Grism Analysis	85
8.4.1	Rerunning the Pipeline	85
8.4.2	Verifying the Output	86

List of Tables

1	List of Acronyms	v
3.1	ODF data file identifiers.	7
3.2	Pipeline Processing groupings.	8
3.3	Pipeline Processing data files	9
5.1	Example datasets used in this Guide.	14
8.1	Some of the important columns in the source list file.	81
8.2	File ID equivalences between repipelined and PPS OM files.	83

Table 1: List of Acronyms

ARF	Ancillary Region File
CAL	Calibration Access Layer
CCD	Charge Coupled Device
CCF	Current Calibration File
CIF	Calibration Index File
EPIC	European Photon Imaging Camera
FITS	Flexible Image Transport System
GO	Guest Observer
GOF	NASA/GSFC Guest Observer Facility
GSFC	Goddard Space Flight Center
GUI	Graphical User Interface
HEASARC	High Energy Astrophysics Science Archive Research Center
HTML	Hyper Text Markup Language
OAL	ODF Access Layer
ODF	Observation Data File
OM	Optical Monitor
PDF	Portable Data Format
PP	Pipeline Processing System
PPS	Pipeline Processing
PV	Performance Validation
RGS	Reflection Grating Spectrometer
RMF	Redistribution Matrix File
SAS	Science Analysis System
SOC	Science Operations Center
SSC	Survey Science Centre
SV	Science Validation
XMM	X-ray Multi-Mirror Mission

Chapter 1

Introduction

The purpose of this *ABC Guide to XMM-Newton* data analysis is to provide a simple walk-through of basic data extraction and analysis tasks. Also included is a guide to references and help available to aid in the analysis of the data. We have tried to balance providing enough information to give the user a useful introduction to a variety of analysis tasks with not providing too much information, which would make a guide like this too ponderous to use. As such, there is no intention to replace the SAS Handbook, which should be considered the highest authority for the use of SAS. Therefore this document will not display the full versatility of the SAS tasks, and of SAS itself, but it will hopefully show a path through the forest.

Chapter 2 provides lists of web-based references for the *XMM-Newton* project, help desks, analysis guides, and science and calibration data. Chapter 3 provides a description of the data files provided for observation data sets. Chapter 4 discusses the installation and use of SAS. Chapters 6, 7, and 8 discuss the analysis of EPIC, RGS, and OM data respectively.

This document will continue to evolve. Updated versions will be made available on our web site at:
<http://heasarc.gsfc.nasa.gov/docs/xmm/abc/>

1.1 ACKNOWLEDGMENTS

This guide would not have been possible without the help and comments from all people involved in the *XMM-Newton* project. In particular, we would like to thank Giuseppe Vacanti and Julian Osborne whose comments made this a more complete and accurate document.

IMH wishes to thank all the OM calibration team and in particular Antonio Talavera, Matteo Guainazzi and Bing Chen for their help in the preparation of this and other documents related to the OM.

SLS wishes to thank Dave Lumb, Richard Saxton, and Steve Sembay for their helpful insights into EPIC data analysis.

Chapter 2

Useful Information and References

2.1 MAIN WEB SITES

- *XMM-Newton* SOC, fount of all *XMM-Newton* project information:

<http://xmm.esac.esa.int/>

- NASA/GSFC GOF, source of US specific information and a mirror site for software and public data access:

<http://xmm.gsfc.nasa.gov/>

- Survey Science Centre

<http://xmmssc-www.star.le.ac.uk/>

2.2 *XMM-NEWTON* HELP DESKS

- The main project helpdesk is located at Vilspa and can be accessed through the WWW:

http://xmm.esac.esa.int/external/xmm_user_support/helpdesk.shtml

or via e-mail:

xmmhelp@sciops.esa.int

The helpdesk also provides an archive of previously asked questions.

- The NASA/GSFC GOF offers an e-mail helpdesk for both general support and for US-specific issues:

xmmhelp@lists.nasa.gov

Some questions addressed to the NASA/GSFC GOF may be redirected to the Vilspa helpdesk.

2.3 MISSION PLANNING AND SPACECRAFT STATUS

- Observation Log:

http://xmm2.esac.esa.int/external/xmm_mission_plan/index.php

The scheduling information from this data base has been extracted and incorporated into a Browse data base at GSFC:

<http://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl>

- Long-Term Timeline:

http://xmm.esac.esa.int/external/xmm_sched/advance_plan.shtml

2.4 PUBLIC DATA ARCHIVES

- SOC Public Data Archive via the XSA:

<http://xmm.esac.esa.int/xsa/>

- GSFC Archive Mirror Site via Browse:

<http://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl>

2.5 CALIBRATION DATA

- *XMM-Newton* Calibration Page. Under this page can be found the Current Calibration File (CCF) archive, release notes for CCF updates, EPIC response and background files (top menu), and calibration information.

http://xmm.esac.esa.int/external/xmm_calibration/

- Caldb, NASA/GSFC GOF mirror site for canned response files:

<ftp://legacy.gsfc.nasa.gov/caldb/data/xmm/>

2.6 SOFTWARE

- *XMM-Newton* Standard Analysis System (SAS):

http://xmm.esac.esa.int/external/xmm_data_analysis/

- HEASARC HEASoft Package:

<http://heasarc.gsfc.nasa.gov/docs/corp/software.html>

- CXC CIAO Package:

<http://asc.harvard.edu/ciao/>

2.7 ANALYSIS, DOCUMENTATION AND HELPFUL HINTS

- On-Line SAS Handbook:

<http://xmm.esac.esa.int/sas/current/howtousesas.shtml>

click “SAS User Guide”

- There is a “watchout” page for current SAS bugs at:

<http://xmm.esac.esa.int/sas/current/watchout/>

- *XMM-Newton* Users Handbook:

http://xmm.esac.esa.int/external/xmm_user_support/documentation/uhb_frame.shtml

- This Guide:

<http://heasarc.gsfc.nasa.gov/docs/xmm/abc/>

- The MPE Analysis Guide:

<http://www.mpe.mpg.de/xray/wave/xmm/cookbook/>

- The Birmingham Analysis Guide (scripts etc. for EPIC extended source analysis):

<http://www.sr.bham.ac.uk/xmm2/>

Chapter 3

Data

3.1 USEFUL DOCUMENTATION

There are a number of documents which the users of *XMM-Newton* data should be aware of. These documents include the *SSC Products Specification*, *Data Files Handbook*, *Reading Data Products CD's* (the most recent versions of these documents can be found in the SOC Document section under

http://xmm.esac.esa.int/external/xmm_user_support/documentation/uhb_frame.shtml), and the *SAS Users Guide* (http://xmm.esac.esa.int/external/xmm_user_support/documentation/sas_usg/USG/).

For observation data sets going to US PIs, the GOF makes the data available online after PGP encryption and after converting the file names to upper case. When the proprietary period for the observation expires the data are decrypted leaving the file names unchanged. A simple decryption script, minus the relevant keys of course, can be found at:

<ftp://legacy.gsfc.nasa.gov/xmm/software/decrypt.pl>.

NOTE: Laura Brenneman wrote a script and accompanying help file that gives explicit directions on how to most quickly pull over all the files in a data set from the archive, as well as decrypting, and uncompressing the files in preparation for data analysis. This package can be found at:

ftp://legacy.gsfc.nasa.gov/xmm/software/prepare_xmm_data.tar.gz. and contains the following files: README, decrypt.pl, and prepare_xmm_data.pl.

3.2 THE DATA

One of the first steps that should be taken when examining your data is to check to see what you actually have. *XMM-Newton* observations can be broken into several exposures which are each assigned separate observation numbers. These separate exposures can be radically different in length and can also have the different instruments in different modes. For example, in one case the full observation was 60 ks with EPIC and RGS active but there was one delivered exposure which was ~ 3 ks and had only RGS active. (This can happen because the RGS can operate farther into regions of higher radiation than the EPIC detector. The additional observation time can be considered an additional exposure with only the RGS active.) Two files are useful for this examination. First, the primary HTML page is `INDEX.HTM` which is included in the Pipeline Products. This page lists basic information for the observation plus the operational modes, filters, and exposure start and stop times for the individual instruments. It also has links to various summary pages, including those for the instruments. (In the case above, the EPIC summary page simply stated that “EPIC exposures processed by PPS None.”) Specifically, **LOOK** at the `P*SUMMAR0000.HTM` files in the pipeline products (easily available through the links). Second, to quickly access images from the various instruments, examine the PPSGRA Pipeline Products page, viewable with a web browser. These files have the nomenclature `P*PPSGRA0000.HTM`. (see § 3.3.3).

3.3 PI Data

Proprietary *XMM-Newton* data is available for download via your XSA account. Email instructions from the XMM-SOC at Vilspa are sent to the address on record with detailed directions on how to retrieve your data via the XSA.

The data files can be considered to come in two groups in separate subdirectories when retrieved, the Observation Data Files (ODF) files and Pipeline Processing (PPS) files. The ODF data contain all of the observation-specific data necessary for reprocessing the observation. The PPS data contain, among other things, calibrated photon event files and source lists.

For observation data sets going to US PIs, the GSFC GOF makes the data available in two directories containing the following groups of files.

- ODF – The ODF (raw) data files
- PPS – The pipeline processed data products

3.3.1 ODF Data

ODF data come with file names in the following format:

- `mmmm_iiiiijjkk_aabeeccfff.zzz`

`mmmm` – revolution orbit number

`iiiiii` – proposal number

`jj` – target ID number in proposal

`kk` – exposure number for target

(**NOTE:** The ten-digit combination of `iiiiijjkk` is the **observation number** and is used repetitively throughout the file nomenclature.)

`aa` – detector (M1 – MOS1, M2 – MOS2, PN – PN, OM – OM, R1 – RGS1, R2 – RGS2, SC – spacecraft)

`b` – flag for scheduled (S) or unscheduled (U) observations, or general purpose X files

`eee` – exposure number

`cc` – CCD number or OM window number

`fff` – data identifier for the three detectors or spacecraft itself; see Table 3.1.

`zzz` – format (FIT - FITS, ASC - ASCII)

NOTE: For SAS processing, the file names should contain all upper case characters. However, at least with early CDs, the file names used lower case characters. The GSFC *XMM-Newton* GOF provides a script to rename the files.

3.3.2 Pipeline Product Data – Summary Files and Groupings

The pipeline processing produces quite a number of useful products which allow a first look at the data, but can overwhelm the user by their sheer numbers. The first place to look is the `INDEX.HTM` page which organizes the presentation of the data and provides links to other PP pages. The `INDEX.HTM` page also lists general observation information (target, date, time, etc.) and instrument modes.

The `INDEX.HTM` page provides links to various observation summary pages, which have names with the following nomenclature:

- `PPiiiiijjkkAAAX000SUMMAR0000.HTM`

`iiiiijjkk` – observation number

`AA` – detector ID (EP - EPIC, OM - Optical Monitor, RG - RGS, OB - Observation)

PP data contain some immediately useful data products such as calibrated photon event lists, source lists, and images. While there are a large number of products which come in a single directory, they can be associated in up to 15 groupings; see Table 3.2. (The number of groups can vary depending on the number of operational instruments, e.g., if the OM is turned off there are no OM products.) Further information on each of these groupings and associated files, such as file contents, file types, and how they may be viewed, can be found in Table 3.3. Each group has an associated HTML file which organizes access to the files and provides a limited description of them. The names of the HTML files are of the following form:

- `PPiiiiijjkkAAAAA000_0.HTM`

`iiiiijjkk` – observation number

`AAAAA` – group identifier (see Table 3.2)

3.3.3 Pipeline Product Data – Data Files

The data file names are of the form (see Table 41 in the *XMM Data Files Handbook*):

- PiiiiijjkkkaablllCCCCCnmmm.zzz

iiiiijjkk – observation number

aa – detector, M1 - MOS1, M2 - MOS2, PN - PN, CA - for files from the CRSCOR group, R1 - RGS1, R2 - RGS2, OM - OM.

b – flag for scheduled (S) or unscheduled (U) observations, or X for files from the CRSCOR group (and any product that is not due to a single exposure)

lll – exposure number

CCCCC – file identification for data from each detector; see Table 3.3

n – For EPIC data, this is the exposure map band number; for RGS data, this is the spectral order number; for the OM, this is the OM window within the exposure.

mmm – source number in hexadecimal

zzz – file type (e.g., PDF, PNG, FTZ, HTM)

ASC - ASCII file, use a web browser, or the “more” command

ASZ - gzipped ASCII file

FTZ - gzipped FITS format, use *ds9*, *Ximage*, *Xselect*, *fv*

HTM - HTML file, use Firefox or other web browser

PDF - Portable Data Format, use *Acrobat Reader*

PNG - Portable Networks Graphics file, use a web browser

TAR - TAR file

Table 3.1: ODF data file identifiers.

Data ID	Contents
EPIC files	
IME	Event list for individual CCDs, imaging mode
RIE	Event list for individual CCDs, reduced imaging mode
CTE	Event list for individual CCDs, compressed timing mode
TIE	Event list for individual CCDs, timing mode
BUE	Event list for individual CCDs, burst mode
AUX	Auxiliary file
CCX	Counting cycle report (auxiliary file)
HBH	HBR buffer size, non-periodic housekeeping
HCH	HBR configuration, non-periodic housekeeping
HTH	HBR threshold values, non-periodic housekeeping
PEH	Periodic housekeeping
PTH	Bright pixel table, non-periodic housekeeping
DLI	Discarded lines data
PAH	Additional periodic housekeeping
PMH	Main periodic housekeeping
RGS files	
AUX	on-board processing statistics
SPE	raw event list for one CCD
DII	diagnostic images
D1H	CCD readout settings
D2H	CCD readout settings
PFH	housekeeping data
ODX	pixel offset data
XMM files	
ATS	spacecraft attitude history
OM files	
IMI	imaging file
THX	tracking history file
WDX	window data auxiliary file
NPH	non-periodic housekeeping file
PEH	periodic housekeeping file
PAX	field acquisition data
RFX	priority reference frame data
PFX	priority fast mode data
FAE	event list (if fast mode was used)

Table 3.2: Pipeline Processing groupings.

Group ID	Contents
PP files	
PPSDAT	Contains the Calibration Index File (CIF) used in the pipeline processing (*CALIND*), PPS information, and the attitude history time series (*ATTTSR*) in gzipped FITS or ASCII format.
PPSGRA	Contains the OM tracking history plots, PPS, EPIC, OM, RGS observation, and PPS run summaries. NOTE: CHECK THESE OUT
PPSMMSG	ASCII file containing pipeline processing report
EPIC files	
CRSCOR	Contains PDF files of POSS II finding charts, HTML files of cross correlations with the SIMBAD data base, FITS tables for the detected sources
EANCIL	Contains the exposure maps in a variety of energy bands and the source-detection sensitivity maps for the EPIC instruments. The sensitivities are in units of counts s ⁻¹ corrected for vignetting and corresponding to a likelihood specified in the FITS header. The files are gzipped with a .FTZ extension.
EEVLIS	Contains calibrated photon event files for the EPIC detectors. If the files are sufficiently large they may be separated into two tar files. The files are gzipped fits files with a .FTZ extension.
ESKYIM	This group contains the event images in a variety of energy bands. The fits files are gzipped with a .FTZ extension, the full images also come as PNG images.
ESRLIS	Contains EPIC observation source lists. There is an HTML page of the merged source list and gzipped fits tables of source lists from the different instruments and source detection tasks.
OM files	
OIMAGE	Contains OM sky images in gzipped FITS format.
OMSLIS	Contains OM observation source lists in gzipped FITS format.
OMSRTS	Contains OM star tracking time series in gzipped FITS format.
RGS files	
REVLIS	Contains the RGS source and event lists in gzipped FITS format
REXPIM	Contains the RGS exposure maps in gzipped FITS format
RIMAGE	Contains the RGS images (both energy dispersion and cross dispersion) in gzipped FITS and PNG formats
RSPECT	Contains the RGS source and background spectra in gzipped FITS and PDF formats

Table 3.3: Pipeline Processing data files

Group ID	File ID	Contents	File Type	View With
EPIC files				
CRSCOR	FCHART	Finding chart	PDF	<i>Acrobat Reader</i>
	ROSMG	ROSAT image of region	PDF	<i>Acrobat Reader</i>
	SNNNN ¹	Source cross-correlation results	Zipped FITS	<i>fv</i>
	DNNNN ¹	Catalog descriptions	PDF	<i>Acrobat Reader</i>
	FNNNN ¹	FOV cross-correlation result	Zipped FITS	<i>fv</i>
ESKYIM	IMAGE_8	Sky image 0.2 - 12.0 keV	Zipped FITS	<i>ds9, Ximage, fv</i>
	IMAGE_1	Sky image 0.2 - 0.5 keV	Zipped FITS	<i>ds9, Ximage, fv</i>
	IMAGE_2	Sky image 0.5 - 2.0 keV	Zipped FITS	<i>ds9, Ximage, fv</i>
	IMAGE_3	Sky image 2.0 - 4.5 keV	Zipped FITS	<i>ds9, Ximage, fv</i>
	IMAGE_4	Sky image 4.5 - 7.5 keV	Zipped FITS	<i>ds9, Ximage, fv</i>
	IMAGE_5	Sky image 7.5 - 12.0 keV	Zipped FITS	<i>ds9, Ximage, fv</i>
EANCIL	EXPMAP_8	Exposure map 0.2 - 12.0 keV	Zipped FITS, PNG	<i>ds9, Ximage, fv, web browser</i>
	EXPMAP_1	Exposure map 0.2 - 0.5 keV	Zipped FITS	<i>ds9, Ximage, fv</i>
	EXPMAP_2	Exposure map 0.5 - 2.0 keV	Zipped FITS	<i>ds9, Ximage, fv</i>
	EXPMAP_3	Exposure map 2.0 - 4.5 keV	Zipped FITS	<i>ds9, Ximage, fv</i>
	EXPMAP_4	Exposure map 4.5 - 7.5 keV	Zipped FITS	<i>ds9, Ximage, fv</i>
	EXPMAP_5	Exposure map 7.5 - 12.0 keV	Zipped FITS	<i>ds9, Ximage, fv</i>
EEVLIS ²	MIEVLI	MOS imaging mode event list	Zipped FITS	<i>xmmselect, fv, Xselect</i>
	PIEVLI	PN imaging mode event list	Zipped FITS	<i>xmmselect, fv, Xselect</i>
	TIEVLI	PN, MOS timing mode event list	Zipped FITS	<i>xmmselect, fv, Xselect</i>
ESRLIS	EBLSLI	Box-local detect source list	Zipped FITS	<i>fv</i>
	EBMSLI	Box-map detect source list	Zipped FITS	<i>fv</i>
	EMSRLI	Max-like detect source list	Zipped FITS	<i>fv</i>
	OBSMLI	Summary source list	Zipped FITS, HTML	<i>fv, web browser</i>
RGS files				
REVLIS	SRCLI_	RGS Source Lists	Zipped FITS	<i>fv</i>
	EVENLI	RGS Event lists	Zipped FITS	<i>xmmselect, fv</i>
REXPIM	EXPMAP	RGS Exposure Maps	Zipped FITS	<i>ds9, Ximage, fv</i>
RSPECT	SRSPEC1	1st Order Source Spectra	Zipped FITS	<i>Xspec, fv</i>
	SRSPEC2	2nd Order Source Spectra	Zipped FITS	<i>Xspec, fv</i>
	BGSPEC1	1st Order Background Spectra	Zipped FITS	<i>Xspec, fv</i>
	BGSPEC2	2nd Order Background Spectra	Zipped FITS	<i>Xspec, fv</i>
	SRSPEC	Spectra Plots	PDF format	<i>Acrobat Reader</i>
RIMAGE	ORDIMG	Images, disp. vs. X-disp	Zipped FITS, PNG	<i>ds9, Ximage, fv, web browser</i>
	IMAGE_	Images, disp. vs. PI	Zipped FITS, PNG	<i>ds9, Ximage, fv, web browser</i>
OM files				
OIMAGE	SIMAGE	OM Sky Image	Gzipped FITS	<i>ds9, Ximage, fv</i>
OMSLIS	SWSRLI	OM Source Lists	Zipped FITS	<i>fv</i>
OMSRTS	TSTRTS	Tracing Star Time Series	Zipped FITS	<i>fv</i>

¹ NNNNN - Alphanumeric ID² Files for only those modes which were active will be included

Chapter 4

Setting Up and Running SAS

The Science Analysis Software (SAS, <http://xmm.esac.esa.int/sas/>), developed by the Survey Science Centre (SSC) and Science Operations Centre (SOC), is a suite of about 125 programs and scripts that perform data reduction, extraction, and some analysis of *XMM-Newton* data. The Pipeline Processing System (PPS), comprised of a superset of the SAS suite and Perl scripts, is run at Leicester University (<http://xmmssc-www.star.le.ac.uk/>) to create the basic data products provided to the Guest Observer from the satellite ancillary and science data. SAS is not designed for higher level scientific analysis such as spectral fitting and temporal analysis, but does provide for the creation of detector response files and barycentric corrected event timing information. SAS includes extensive EPIC and OM source-detection software. The SAS product files conform to OGIP FITS standards so any high-level analysis package used in high-energy astrophysics should theoretically be capable of processing *XMM-Newton* data. For example, the HEASoft package, <http://heasarc.gsfc.nasa.gov/docs/corp/software.html>, of the High Energy Astrophysics Science Archive Research Center (HEASARC, <http://heasarc.gsfc.nasa.gov/>) at NASA/GSFC and the CIAO package (<http://asc.harvard.edu/ciao/>) of the Chandra X-ray Observatory Center (<http://chandra.harvard.edu/>) can both be used with *XMM-Newton* data files.

4.1 Installation

The primary guide for the installation of SAS can be found through the SOC at <http://xmm.esac.esa.int/sas/current/installation/>. Because of the complexity of the SAS installation, it is strongly recommended that users download and install the binary executables rather than compiling SAS from source code (which also necessitates the purchase of commercial software). It should also be noted that “optional” components, while not needed for running SAS tasks from the command-line, are critical to running SAS from the GUI. These optional components are listed at the SOC page <http://xmm.esac.esa.int/sas/current/requirements/>.

4.2 Calibration Data

XMM-Newton data reduction and analysis requires extensive calibration data which must be available under a Current Calibration File (CCF) directory. Information on the CCF and instructions for downloading/mirroring the files can be found under the SOC *XMM-Newton* Calibration page (http://xmm2.esac.esa.int/external/xmm_sw_cal/cal) links to the CCF release notes. In addition, background event files and canned spectral response files can be found under http://xmm2.esac.esa.int/external/xmm_sw_cal/background/index.shtml.

4.3 SAS Environmental Parameters and Invocation

There are a few parameters which need to be set for the proper operation of SAS. Two are taken care of by the initialization script, `SAS_DIR` and `SAS_PATH`. These both set the SAS directory path. The remaining parameters, listed below, still need to be set. (The commands should, of course, be modified to be appropriate for your specific setup.)

```

setenv SAS_CCFPATH /full/path/to/CCF      Sets the directory path to the CCF data
setenv SAS_ODF /full/path/to/ODF          Sets the directory path to the ODF data
setenv SAS_CCF /full/path/to/ODF/ccf.cif   Sets the Calibration Index File (CIF) path
                                           and file name

```

Please note that SAS_CCF can also be set after the creation of the ccf.cif file with *cifbuild* (see §5.1). Also, while not necessary to run SAS, the following parameters are useful to know about and should be set.

```

setenv SAS_VERBOSITY 3                    Sets the verbosity, 1 => little, 10 => lot
setenv SAS_SUPPRESS_WARNING 3             Sets the warning level, 1 => little, 10 => lot
setenv SAS_IMAGEVIEWER ds9                Sets the default image viewer; in this case,
                                           it is ds9, but should be set to whatever
                                           the user prefers.)

```

Finally, SAS is invoked by sourcing the script that came with the SAS package:

```

source /full/path/to/xmmsas_20090615_1801/setsas.csh      Initializes SAS

or

source /full/path/to/xmmsas_20090615_1801/setsas.sh        Alternate SAS initialization

```

SAS can now be used on the command line. To invoke the GUI, type:

```
sas &
```

To verify the SAS-specific settings, type `env | grep SAS`.

It is strongly recommended that users include these environmental settings and make an alias to the initialization script in their login shell file (.cshrc, .bashrc, etc.)! It will save a lot of typing and lower the potential for frustration.

SAS tasks can be run equally well from the command line and from the SAS GUI. In this document we will demonstrate the use of some of the more commonly used tasks from both the GUI and command line, although in some instances, we only give command line examples. In these cases, the GUI can still be used – the user need only set the parameters there.

The MPE Analysis Guide, <http://www.mpe.mpg.de/xray/wave/xmm/cookbook/preparation/index.php> demonstrates many of the common tasks using GUIs.

4.3.1 SAS Helpful Hints

Command lines can often be quite long with a variety of parameters. To avoid considerable typing when creating command scripts a feature of the GUI interface can be of assistance. When invoking a task through the GUI a copy of the full command appears in the dialog box, from where it can then be cut and pasted.

There are several useful features of the command-line interface that users should be aware of. 1) If the `dialog` parameter is included in the command line, the task GUI will pop up with all parameters in the command line preset. This allows the use of the GUI interfaces at the task level without having to go through the main SAS GUI. 2) If the `manpage` parameter is included in the command line, the task documentation will pop up in a web browser window. 3) In addition, the command `sashelp doc=sas_task` will pop up a web browser window with the documentation for the task *sas_task* as well.

The command documentation (i.e., the pages brought up by `sashelp doc=sas_task` or `sas_task manpage`) has an Errors section. Common warning messages produced by the tasks and their meanings are listed here. This feature is *very* useful.

4.4 SAS Syntax and Logic

4.4.1 Command Line Syntax

There is some flexibility in command line syntax in SAS. The following are all valid task calls on the command line that result in identical operations:

```
rgsproc withsrc=F
rgsproc withsrc=no
rgsproc withsrc='no'
rgsproc withsrc="no"
rgsproc --withsrc=no
rgsproc --withsrc='no'
rgsproc --withsrc="no"
```

However,

```
rgsproc -withsrc=F
rgsproc -withsrc=no
rgsproc -withsrc='no'
rgsproc -withsrc="no"
```

are not correct syntax.

One format is not “more correct” than another, and the choice of which to use is left to user preference. In this ABC guide we adopt the simplest format, and use no dashes and only single quotation marks only when required, e.g.,

```
rgsproc withsrc=no orders='1 2 3'
```

where, in this case, the quotes provide the task with a list.

4.4.2 Table Syntax

When a task requires the use of a table within a file there are also several valid syntaxes, e.g.,

```
xmmselect table=filtered.fits:EVENTS
xmmselect table="filtered.fits:EVENTS"
xmmselect table=filtered.fits%EVENTS
```

do an identical operation in opening the EVENTS table inside the file `filtered.fits`.

4.4.3 Filtering Logic

Filtering event files requires some command of the SAS logical language which consists of familiar arithmetic and Boolean operators and functions. These, and their syntax, are described within the on-line documentation supplied with the software. Pull up the help document using:

```
sashelp doc=selectlib
```

Chapter 5

Preparing the Data for Processing

Throughout this Primer, data from the HEASARC archive are used to illustrate how to run tasks for each instrument on XMM-Newton; new users are encouraged to use these sample data, though it should be noted that any data from the relevant instrument will do. Information about the example datasets are in Table 5.1.

A number of SAS tasks, including those which rerun the pipeline, require an up-to-date list of calibration files (the CIF file) and extended Observation Data File (ODF) summary file. These are obtained by running the tasks *cifbuild* and *odfingest*, respectively. Because of these dependencies, **it is strongly recommended that users run these tasks**, whether they plan to rerun the pipeline or not.

First, unzip the all gzipped files in the ODF directory. If necessary, rename all files in the ODF directory to upper case. This can be done using the script provided by the NASA/GSFC XMM-Newton GOF.

```
gunzip ODF/*.gz
```

5.1 *cifbuild*

Many SAS tasks require calibration information from the Calibration Access Layer (CAL). Relevant files are accessed from the set of Current Calibration File (CCF) data using a CCF Index File (CIF). A CIF is included in the pipeline products (*CALIND*), but if the CCF has been updated it can be recreated by the user by simply typing

```
cd ODF
cifbuild
```

The task *cifdiff* can be used to compare the new CIF with the old:

```
cifdiff calindex1set=ccf.cif calindex2set=CALIND.FIT
```

where

calindex1set – name of the first file to be compared, in this case the output from the current run of *cifbuild*
calindex2set – name of the second file to be compared, in this case the (renamed) pipeline product file

If the CAL has changed, it is a good idea to repipeline the data using the new CIF. To help determine whether it is reasonable to repipeline the data, the CCF release notes should be examined (see §2.5). CCF files can be downloaded directly from the SOC website see (see §4.2).

To use the updated CIF file in further processing, you will need to reset the environment variable SAS_CCF:

```
setenv SAS_CCF /full/path/to/ODF/ccf.cif
```

Table 5.1: Example datasets used in this Guide.

Instrument	Chapter	ObsID	Object
EPIC (Imaging mode)	6	0123700101	Lockman Hole
EPIC (Timing mode)	6	0403530301	Bol 375
RGS	7	0153950701	Mkn 421
OM (Image mode)	8.2	0123700101	Lockman Hole
OM (Fast mode)	8.3	0411081601	Mkn 421
OM (Grism mode)	8.4	0125320801	BPM 16274

5.2 *odfingest*

The task *odfingest* extends the Observation Data File (ODF) summary file with data extracted from the instrument housekeeping data files and the calibration database. It is only necessary to run it once on any dataset, and will cause problems if it is run a second time. If for some reason *odfingest* must be rerun, you must first delete the earlier file it produced. This file largely follows the naming convention described in §3.3.3, but has `SUM.SAS` appended to it. After running *odfingest*, you will need to reset the environment variable `SAS_ODF` to its output file. To run *odfingest* and reset the environment variable, simply type:

```
odfingest
```

```
setenv SAS_ODF /full/path/to/ODF/full_name_of_*SUM.SAS
```

The user will likely find it useful to alias these environment variable resets in their login shell (`.cshrc`, `.bashrc`, etc.).

The rest of this guide consists of step-by-step examples of how to reprocess and analyze data for the EPIC (§6), RGS (§7), and OM (§8). At the start of each chapter, it is assumed that the user has downloaded the data, initialized SAS, and set the environment variables, as listed in 4.3. It is also assumed that the CIF file is up-to-date, the `*SUM.SAS` file exists, and the appropriate environment variables reset as needed.

Chapter 6

An EPIC Data Processing and Analysis Primer

So, you’ve received an *XMM-Newton* EPIC data set. What are you going to do with it? After checking what the observation consists of (see § 3.2), you should note when the observation was taken. If it is a recent observation, it was likely processed with the most recent calibrations and SAS, and you can immediately start to analyze the Pipeline Processed data. However, if it is more than a year old, it was probably processed with older versions of CCF and SAS prior to archiving, and the pipeline should be rerun to generate event files with the latest calibrations.

As noted in Chapter 4, a variety of analysis packages can be used for the following steps. However, as the SAS was designed for the basic reduction and analysis of *XMM-Newton* data (extraction of spatial, spectral, and temporal data), it will be used here for demonstration purposes (although see § 6.9 for a short tutorial on the use of *Xselect* for data extraction). SAS will be required at any rate for the production of detector response files (RMFs and ARFs) and other observatory-specific requirements. (Although for the simple case of on-axis point sources the canned response files provided by the SOC can be used.)

NOTE: For PN observations with very bright sources, out-of-time events can provide a serious contamination of the image. Out-of-time events occur because the read-out period for the CCDs can be up to $\sim 6.3\%$ of the frame time. Since events that occur during the read-out period can’t be distinguished from others events, they are included in the event files but have invalid locations. For observations with bright sources, this can cause bright stripes in the image along the CCD read-out direction. For a more detailed description of this issue, check: <http://www.mpe.mpg.de/xray/wave/xmm/cookbook/preparation/index.php>

It is **strongly** recommended that you keep all reprocessed data in its own directory! SAS places output files in whichever directory it is in when a task is called. Throughout this primer, it is assumed that the Pipeline Processed data are in the PPS directory, the ODF data (with upper case file names, and uncompressed) are in the directory ODF, the analysis is taking place in the PROC directory, and the CCF data are in the directory CCF.

If your data are recent, you need only to gunzip the files and prepare the data for processing (see §5. Feel free to skip the command line and GUI discussions on repipelining (§6.1 and §6.4, respectively) and proceed to the command line or GUI discussions on filtering (§6.2 and §6.5, respectively). In any case, for simplicity, it is recommended that you change the name of the unzipped event file to something easy to type. For example, an MOS1 event list:

```
cp PPS/PiiiiijjkkM1S111MIEVLI0000.FTZ PROC/mos1.fits
```

where

```
iiiiijjkk – observation number  
111 – exposure number within the observation
```

Various analysis procedures are demonstrated using the Lockman Hole SV1 dataset, ObsID 0123700101, which definitely needs to be repipelined. The following procedures are applicable to all *XMM-Newton* datasets, so it is not required that you use this particular dataset; any observation should be sufficient.

If you simply want to have a quick look at your data, the ESKYIM files contain EPIC sky images in different energy bands whose ranges are listed in Table 3.3. While the zipped FITS files may need to be

unzipped before display in *ds9* (depending on the version of *ds9*), they can be displayed when zipped using *fv* (*fv* is FITS file viewer available in the HEASoft package). In addition, the image of the total band pass for all three EPIC detectors is also provided in PNG format which can be displayed with a web browser. Also, the PP source list is provided in both zipped FITS format (readable by *fv*) and as an HTML file.

For detailed descriptions of PP data nomenclature, file contents, and which tasks can be used to view them, see Tables 3.2 and 3.3. For detailed descriptions of ODF data nomenclature and file contents, see Table 3.1.

The following sections (§6.1, 6.2, and 6.3) will deal with processing EPIC data on the command line. Users who prefer to use the GUI should skip ahead to find the demonstrations for the same tasks with the GUI (§6.4, 6.6, and 6.5).

6.1 Rerun the Pipeline on the Command Line

We assume that the data was prepared and environment variables were set according to §5. In the window where SAS was initialized, in your “processing directory” PROC, run *emchain* or *emproc* to produce calibrated photon event files for the MOS cameras, and *epchain* or *epproc* to do the same for the PN camera.

Note that *emproc* and *epproc* will automatically detect what mode the data were taken in. However, if you are using data that was not taken in Imaging mode and want to run *emchain* or *epchain*, you will need to set the relevant parameter, as shown below.

To process MOS data taken in Imaging Mode (like the Lockman Hole dataset), type

```
emchain
```

or

```
emproc
```

Similarly, to process PN data taken in Imaging Mode, type

```
epchain
```

or

```
epproc
```

If the dataset has more than one exposure, a specific exposure can be accessed using the **exposure** parameter, e.g.:

```
emchain exposure=n
```

where *n* is the exposure number. To create an out-of-time event file for your PN data, add the parameter *withoutoftime* to your *epchain* invocation:

```
epchain withoutoftime=yes
```

If you have PN data taken in Timing or Burst mode, you can either just use *epproc*, or

```
epchain datamode=TIMING
```

and

```
epchain datamode=BURST
```

By default, none of these tasks keep any intermediate files they generate. *Emchain* and *epchain* maintain the naming convention described in §3.3.3. *Emproc* and *epproc* designate their output event files with “Evts.ds”. “*ImagingEvts.ds”, “*TimingEvts.ds”, and “*BurstEvts.ds” denote the imaging mode, timing, and burst mode event lists, respectively. In either case, you may want to name the new files something easy to type. For example, to rename one of the new MOS1 event files output from *emchain* or *emproc*, respectively, type

```
mv P0123700101M1S001MIEVLI0000.FIT mos1.fits
```

or

```
mv 0070_0123700101_EMOS1_S001_ImagingEvts.ds mos1.fits
```

Once the new event files have been obtained, the analysis techniques described below can be used for data in Imaging Mode. For data in Timing Mode, see §6.3.

6.2 Examine and Analyze Data on the Command Line (Imaging Mode)

Remember that all tasks should be called from the window where SAS was initiated, and that tasks place output files in whatever directory you happen to be in when they are called, so be sure to call them in the directory where you want the output to go!

6.2.1 Create and Display an Image

To create an image in sky coordinates, type

```
evselect table=mos1.fits withimageset=yes imageset=image.fits
      xcolumn=X ycolumn=Y imagebinning=imageSize ximagesize=600 yimagesize=600
```

where

```
table - input event table
withimageset - make an image
imageset - name of output image
xcolumn - event column for X axis
ycolumn - event column for Y axis
imagebinning - form of binning, force entire image into a given size or bin by a specified number of pixels
ximagesize - output image pixels in X
yimagesize - output image pixels in Y
```

The output file `image.fits` can be viewed by using a standard FITS display, such as *ds9* (see Figure 6.1) :

```
ds9 image.fits &
```

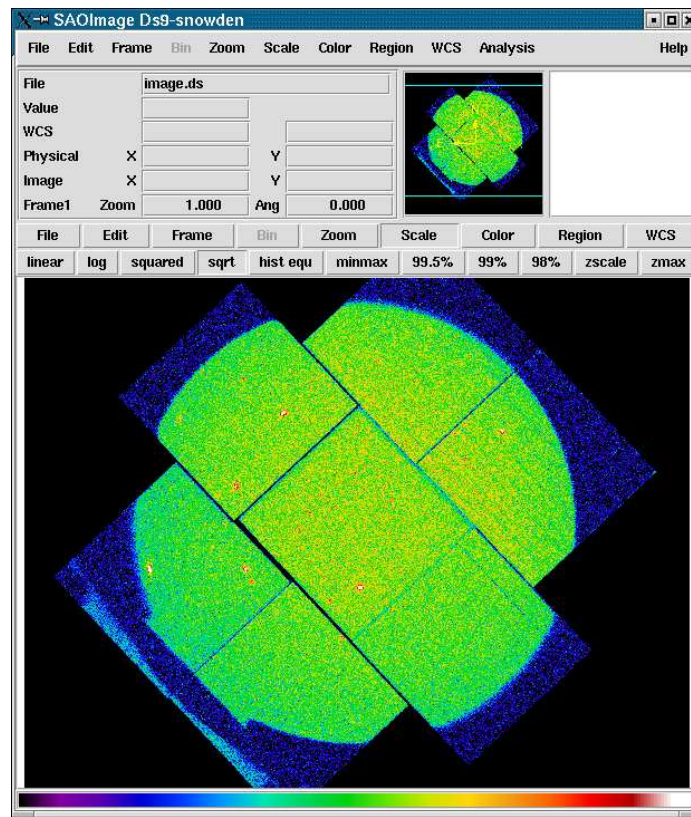



Figure 6.1: The MOS1 image, displayed in *ds9*.

6.2.2 Create and Display a Light Curve

To create a light curve, type

```
evselect table=mos1.fits withrateset=yes rateset=mos1_ltcrv.fits
      maketimecolumn=yes timecolumn=TIME timebinsize=100 makeratecolumn=yes
```

where

```
table – input event table
withrateset – make a light curve
rateset – name of output light curve file
maketimecolumn – control to create a time column
timecolumn – time column label
timebinsize – time binning (seconds)
makeratecolumn – control to create a count rate column, otherwise a count column will be created
```

The output file `mos1_ltcrv.fits` can be viewed by using *fv* or *dsplot*:

```
dsplot table=mos1_ltcrv.fits x=TIME y=RATE.ERROR withoffsetx=yes &
```

where

```
table – input event table
x – column for plotting on X axis
y – column for plotting on Y axis
withoffsetx – creates an offset to the X axis
```

or, alternatively,

```
fv mos1_ltcrv.fits &
```

In the pop-up window, the RATE extension will be available in the second row (index 1, as numbering begins with 0). Select “PLOT” from this row, and select the column name and axis on which to plot it.

6.2.3 Applying Standard Filters the Data

The filtering expressions for the MOS and PN are:

```
(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM
```

and

```
(PATTERN <= 12)&&(PI in [200:15000])&&#XMMEA_EP
```

The first two expressions will select good events with `PATTERN` in the 0 to 12 range. The `PATTERN` value is similar the `GRADE` selection for *ASCA* data, and is related to the number and pattern of the CCD pixels triggered for a given event. The `PATTERN` assignments are: single pixel events: `PATTERN == 0`, double pixel events: `PATTERN in [1:4]`, triple and quadruple events: `PATTERN in [5:12]`.

The second keyword in the expressions, `PI`, selects the preferred pulse height of the event; for the MOS, this should be between 200 and 12000 eV. For the PN, this should be between 200 and 15000 eV. This should clean up the image significantly with most of the rest of the obvious contamination due to low pulse height events. Setting the lower `PI` channel limit somewhat higher (e.g., to 300 eV) will eliminate much of the rest.

Finally, the `#XMMEA_EM` (`#XMMEA_EP` for the PN) filter provides a canned screening set of `FLAG` values for the event. (The `FLAG` value provides a bit encoding of various event conditions, e.g., near hot pixels or outside of the field of view.) Setting `FLAG == 0` in the selection expression provides the most conservative screening criteria and should always be used when serious spectral analysis is to be done on the PN. It typically is not necessary for the MOS.

It is a good idea to keep the output filtered event files and use them in your analyses, as opposed to re-filtering the original file with every task. This will save much time and computer memory. As an example, the

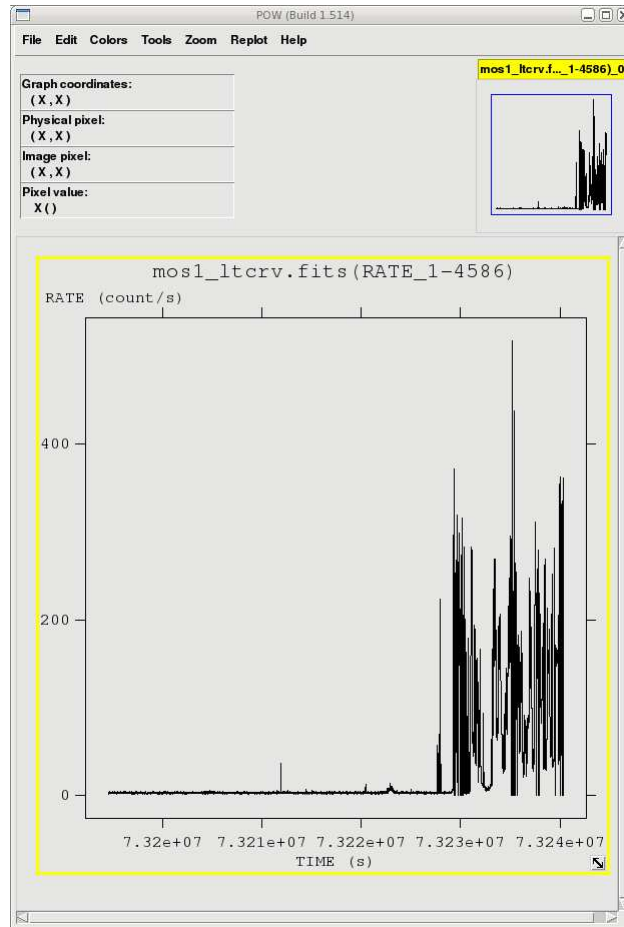


Figure 6.2: The light curve, displayed in fv.

Lockman Hole data's original event file is 48.4 Mb; the fully filtered list (that is, filtered spatially, temporally, and spectrally) is only 4.0Mb!

To filter the data, type

```
evselect table=mos1.fits withfilteredset=yes
  expression='(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM'
  filteredset=mos1_filt.fits filtertype=expression keepfilteroutput=yes
  updateexposure=yes filterexposure=yes
```

where

```
table - input event table
filtertype - method of filtering
expression - filtering expression
withfilteredset - create a filtered set
filteredset - output file name
keepfilteroutput - save the filtered output
updateexposure - for use with temporal filtering
filterexposure - for use with temporal filtering
```

6.2.4 Applying Time Filters the Data

Sometimes, it is necessary to use filters on time in addition to those mentioned above. This is because of soft proton background flaring, which can have count rates of 100 counts/sec or higher.

It should be noted that the amount of flaring that needs to be removed depends in part on the object observed; a faint, extended object will be more affected than a very bright X-ray source.

There are two ways to filter on time: with an explicit reference to the `TIME` or `RATE` parameters in the filtering expression, or by creating a secondary Good Time Interval (GTI) file with the task *tabgtigen*. Both procedures are described below. For the example data, we will filter by time, though you can just as easily filter by rate.

To explicitly define the `TIME` or `RATE` parameters, make a light curve and display it, as demonstrated in §6.2.2 and plotted in Figure 6.2. There is a very large flare toward the end of the observation, so the syntax for the time selection is `(TIME <= 73227600)`. However, there is also a small flare within an otherwise good interval. A slightly more complicated expression to remove it would be: `(TIME <= 73227600)&&!(TIME IN [73221920:73223800])`. The syntax `&&(TIME < 73227600)` includes only events with times less than 73227600, and the “!” symbol stands for the logical “not”, so use `&&!(TIME IN [73221920:73223800])` to exclude events in the time interval 73221920 to 73223800.

If combined with the standard filtering expression (see §6.2.3), the full filtering expression would then be:

```
(PATTERN <= 12)&&(PI IN [200:12000])&&#XMMEA_EM
&&(TIME <= 73227600) &&!(TIME IN [73221920:73223800])
```

This expression can then be used to filter the original event file, as shown in §6.2.3, or only the times can be used to filter the file that has already had the standard filters applied:

```
evselect table=mos1_filt.fits withfilteredset=yes
expression='(TIME <= 73227600) &&!(TIME IN [73221920:73223800])'
filteredset=mos1_filt_time.fits filtertype=expression keepfilteroutput=yes
updateexposure=yes filterexposure=yes
```

where the keywords are as described in §6.2.3.

To filter on time using a secondary GTI file, make the file by using the same time filtering parameters as determined above and the *tabgtigen* task,

```
tabgtigen table=mos1_ltcrv.fits gtiset=gtiset.fits timecolumn=TIME
expression='(TIME <= 73227600)&&!(TIME IN [73221920:73223800])'
```

where

```
table – input count rate table
expression – filtering expression
gtiset – output file name for selected GTI intervals
timecolumn – time column
```

and apply the new GTI file with the *evselect* task:

```
evselect table=mos1_filt.fits withfilteredset=yes
expression='GTI(gtiset.fits,TIME)' filteredset=mos1_filt_time.fits
filtertype=expression keepfilteroutput=yes
updateexposure=yes filterexposure=yes
```

where

```
table – input count rate table
expression – filtering expression
withfilteredset – create a filtered set
filteredset – output file name
filtertype – method of filtering
keepfilteroutput – save the filtered set
updateexposure – update exposure information in event list and in spectrum files
filterexposure – filter exposure extensions of event list with same time
filters as for corresponding CCD
```

6.2.5 Source Detection with *edetect_chain*

The *edetect_chain* task does nearly all the work involved with EPIC source detection, but it requires some input files that must be generated first. The first step is to make an attitude file with the task *atthkgen*, then images of the sources in the desired energy bands, with the image binning sizes as needed according to the detector. For the MOS, the binsize is 22; for the PN, the binsize is 82.

In the example below, source detection is done only on MOS1 images in two bands (“soft” X-rays with energies between 300 and 2000 eV, and “hard” X-rays, with energies between 2000 and 10000 eV).

The example uses the filtered event file produced in §6.2.4, with the assumption that it is located in the current directory.

First, make the attitude file by typing

```
atthkgen atthkset=attitude.fits timestep=1
```

where

```
atthkset – output file name
timestep – time step in seconds for attitude file
```

Next, make the soft and hard X-ray images with *evselect* by typing

```
evselect table=mos1_filt_time.fits withimageset=yes imageset=mos1-s.fits
imagebinning=binSize xcolumn=X ximagebinsize=22 ycolumn=Y yimagebinsize=22
filtertype=expression expression='(FLAG == 0)&&(PI in [300:2000])'
```

and

```
evselect table=mos1_filt_time.fits withimageset=yes imageset=mos1-h.fits
imagebinning=binSize xcolumn=X ximagebinsize=22 ycolumn=Y yimagebinsize=22
filtertype=expression expression='(FLAG == 0)&&(PI in [2000:10000])'
```

We will also make an image with both soft and hard X-rays for display purposes:

```
evselect table=mos1_filt_time.fits withimageset=yes imageset=mos1-all.fits
imagebinning=binSize xcolumn=X ximagebinsize=22 ycolumn=Y yimagebinsize=22
filtertype=expression expression='(FLAG == 0)&&(PI in [300:10000])'
```

where the parameters are

```
table – event list
withimageset – flag to create an image
imageset – fits image name to be created
imagebinning – how to bin the image
xcolumn – table column to use for the X axis
ximagebinsize – binning in X axis
ycolumn – table column to use for the Y axis
yimagebinsize – binning in Y axis
filtertype – type of filtering
expression – filtering expression
```

Now we can run *edetect_chain*:

```
edetect_chain imagesets='mos1-s.fits mos1-h.fits'
eventsets='mos1_filt_time.fits' attitudeset=attitude.fits
pimin='300 2000' pimax='2000 10000' likemin=10 witheexpmap=yes
ecf='0.878 0.220' eboxl_list=eboxlist_l.fits
eboxm_list=eboxlist_m.fits eml_list=emllist.fits esp_withootset=no
```

where

```

imagesets – list of count images
eventsets – list of event files
attitudeset – attitude file name
pimin – list of minimum PI channels for the bands
pimax – list of maximum PI channels for the bands
likemin – maximum likelihood threshold
withexpmap – create and use exposure maps
ecf – energy conversion factors for the bands
eboxl_list – output file name for the local sliding box source
    detection list
eboxm_list – output file name for the sliding box source detection in
    background map mode list
eml_list – output file name for maximum likelihood source detection list
esp_withootset – Flag to use an out-of-time processed PN event file,
    useful in cases where bright point sources have left streaks in the PN data
esp_ooteventset – The out-of-time processed PN event file

```

The energy conversion factors (ECFs) convert the source count rates into fluxes. The ECFs for each detector and energy band depend on the pattern selection and filter used during the observation. For more information, please consult the calibration paper “SSC-LUX-TN-0059”, available at the XMM-Newton Science Operations Center or see Table 8 in the 3XMM Catalogue User Guide. Those used here are derived from PIMMS using the flux in the 0.1-10.0 keV band, a source power-law index of 1.9, an absorption of 0.5×10^{20} .

We can display the results of *eboxdetect* using the task *srcdisplay* and produce a region file for the sources.

```

srcdisplay boxlistset=emllist.fits imageset=mos1-all.fits
    regionfile=regionfile.txt sourceradius=0.01 withregionfile=yes

```

where

```

boxlistset – eboxdetect source list
imageset – image file name over which the source circles are to be plotted
includesources – flag to include the source positions on the display
regionfile – file name of output file containing source regions
sourceradius – radius of circle plotted to locate sources
withregionfile – flag to create a region file

```

Figure 6.3 shows the MOS1 image overlayed with the detected sources.

6.2.6 Extract the Source and Background Spectra

Throughout the following, please keep in mind that some parameters are instrument-dependent. The parameter **specchannelmax** should be set to 11999 for the MOS, or 20479 for the PN. Also, for the PN, the most stringent filters, **(FLAG==0)&&(PATTERN<=4)**, must be included in the **expression** to get a high-quality spectrum.

For the MOS, the standard filters should be appropriate for many cases, though there are some instances where tightening the selection requirements might be needed. For example, if obtaining the best-possible spectral resolution is critical to your work, and the corresponding loss of counts is not important, only the single pixel events should be selected (**PATTERN==0**). If your observation is of a bright source, you again might want to select only the single pixel events to mitigate pile up (see §6.2.8 and §6.2.9 for a more detailed discussion).

In any case, you’ll need to know spatial information about the area over which you want to extract the spectrum, so display the filtered event file with *ds9*:

```

ds9 mos1_filt_time.fits &

```

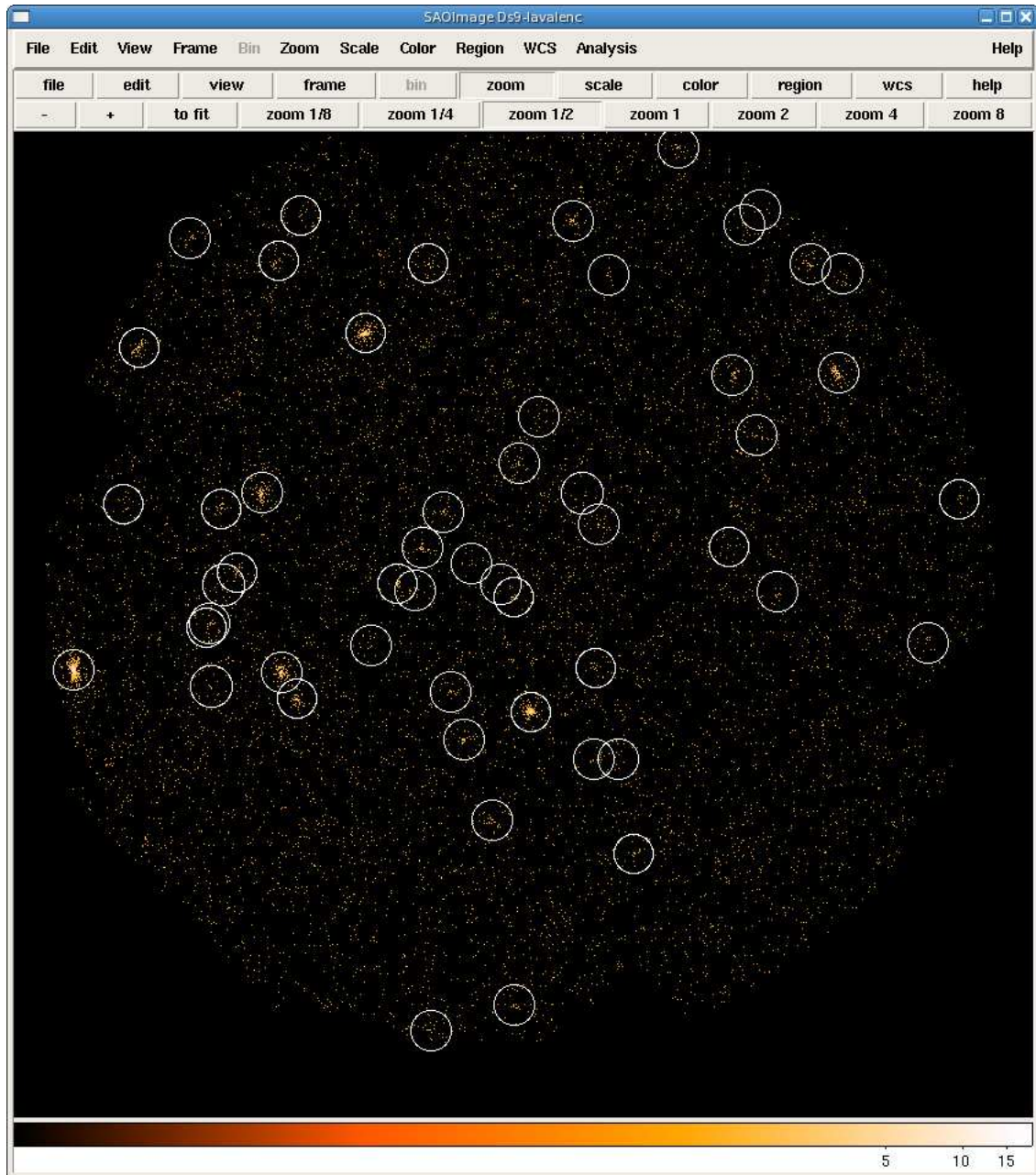


Figure 6.3: MOS1 count image with the detected sources.

Click on the object whose spectrum you wish to extract. This will produce a circle (extraction region), centered on the object. The circle's radius can be changed by clicking on it and dragging to the desired size. Adjust the size and position of the circle until you are satisfied with the extraction region; then, double-click on the region to bring up a window showing the center coordinates and radius of the circle. For this example, we will choose the source at (26188.5,22816.5) and set the extraction radius to 300.

To extract the source spectrum, type

```
evselect table='mos1_filt_time.fits' energycolumn='PI' withfilteredset=yes
  filteredset='mos1_filtered.fits' keepfilteroutput=yes filtertype='expression'
  expression='((X,Y) in CIRCLE(26188.5,22816.5,300))'
  withspectrumset=yes spectrumset='mos1_pi.fits' spectralbinsize=5
```

```
withspecranges=yes specchannelmin=0 specchannelmax=11999
```

where

```
table      - the event file
energycolumn - energy column
withfilteredset - make a filtered event file
keepfilteroutput - keep the filtered file
filteredset - name of output file
filtertype - type of filter
expression - expression to filter by
withspectrumset - make a spectrum
spectrumset - name of output spectrum
spectralbinsize - size of bin, in eV
withspecranges - covering a certain spectral range
specchannelmin - minimum of spectral range
specchannelmax - maximum of spectral range
```

When extracting the background spectrum, follow the same procedures, but change the extraction area. For example, make an annulus around the source; this can be done using two circles, each defining the inner and outer edges of the annulus, then change the filtering expression (and output file name) as necessary.

To extract the background spectrum, type

```
evselect table=mos1_filt_time.fits energycolumn='PI' withfilteredset=yes
filteredset='bkg_filtered.fits' keepfilteroutput=yes filtertype='expression'
expression='((X,Y) in CIRCLE(26188.5,22816.5,1500))&&!((X,Y) in CIRCLE(26188.5,22816.5,500))'
withspectrumset=yes spectrumset='bkg_pi.fits' spectralbinsize=5
withspecranges=yes specchannelmin=0 specchannelmax=11999
```

where the keywords are as described above.

6.2.7 Determine the Spectrum Extraction Areas

The source and background region areas can now be found. This is done with the task *backscale*, which takes into account any bad pixels or chip gaps, and writes the result into the BACKSCAL keyword of the spectrum table.

To find the source and background extraction areas, type

```
backscale spectrumset=mos1_pi.fits badpixlocation=mos1_filt_time.fits
```

and

```
backscale spectrumset=bkg_pi.fits badpixlocation=mos1_filt_time.fits
```

6.2.8 Check for Pile Up

Depending on how bright the source is and what modes the EPIC detectors are in, event pile up may be a problem. Pile up occurs when a source is so bright that incoming X-rays strike two neighboring pixels or the same pixel in the CCD more than once in a read-out cycle. In such cases the energies of the two events are in effect added together to form one event. If this happens sufficiently often, 1) the spectrum will appear to be harder than it actually is, and 2) the count rate will be underestimated, since multiple events will be undercounted. To check whether pile up may be a problem, use the SAS task *epatplot*. (Heavily piled sources will be immediately obvious, as they will have a “hole” in the center.) Note that this procedure requires as input the event files created when the spectrum was made.

The output of *epatplot* is a postscript file, which may be viewed with viewers such as *gv*, containing two graphs describing the distribution of counts as a function of PI channel; see Figure 6.4.

A few words about interpreting the plots are in order. The top is the distribution of counts versus PI channel for each pattern class (single, double, triple, quadruple), and the bottom is the expected pattern distribution (**smooth lines**) plotted over the observed distribution (**histogram**). The lower plot shows the model distributions for single and double events and the observed distributions. It also gives the ratio of observed-to-modeled events with $1\text{-}\sigma$ uncertainties for single and double pattern events over a given energy range. (The default is 0.5-2.0 keV; this can be changed with the `pileupnumberenergyrange` parameter.) If the data is not piled up, there will be good agreement between the modeled and observed single and double event pattern distributions. Also, the observed-to-modeled fractions for both singles and doubles in the 0.5-2.0 keV range will be unity, within errors. In contrast, if the data is piled up, there will be clear divergence between the modeled and observed pattern distributions, and the observed-to-modeled fraction for singles will be less than 1.0, and for doubles, it will be greater than 1.0.

Finally, when examining the plots, it should be noted that the observed-to-modeled fractions can be inaccurate. Therefore, the agreement between the modeled and observed single and double event pattern distributions should be the main factor in determining if an observation is affected by pile up or not.

The source used in our Lockman Hole example is too faint to provide reasonable statistics for *epatplot* and is far from being affected by pile up. An example of a bright source (from a different observation) which is strongly affected by pileup is shown in Figure 6.5. Note that the observed-to-model fraction for doubles is over 1.0, and there is severe divergence between the model and the observed pattern distribution.

To check for pile up in our Lockman Hole example, type

```
epatplot set=mos1_filtered.fits plotfile=mos1_epat.ps useplotfile=yes
        withbackgroundset=yes backgroundset=bkg_filtered.fits
```

6.2.9 My Observation is Piled Up! Now What?

There are a few ways to deal with pile up. First, using the region selection and event file filtering procedures demonstrated in earlier sections, you can excise the inner-most regions of a source (as they are the most heavily piled up), re-extract the spectrum, and continue your analysis on the excised event file. For this procedure, it is recommended that you take an iterative approach: remove an inner region, extract a spectrum, check with *epatplot*, and repeat, each time removing a slightly larger region, until the model and observed distribution functions agree.

You can also use the event file filtering procedures to include only single pixel events (`PATTERN==0`), as these events are less sensitive to pile up than other patterns.

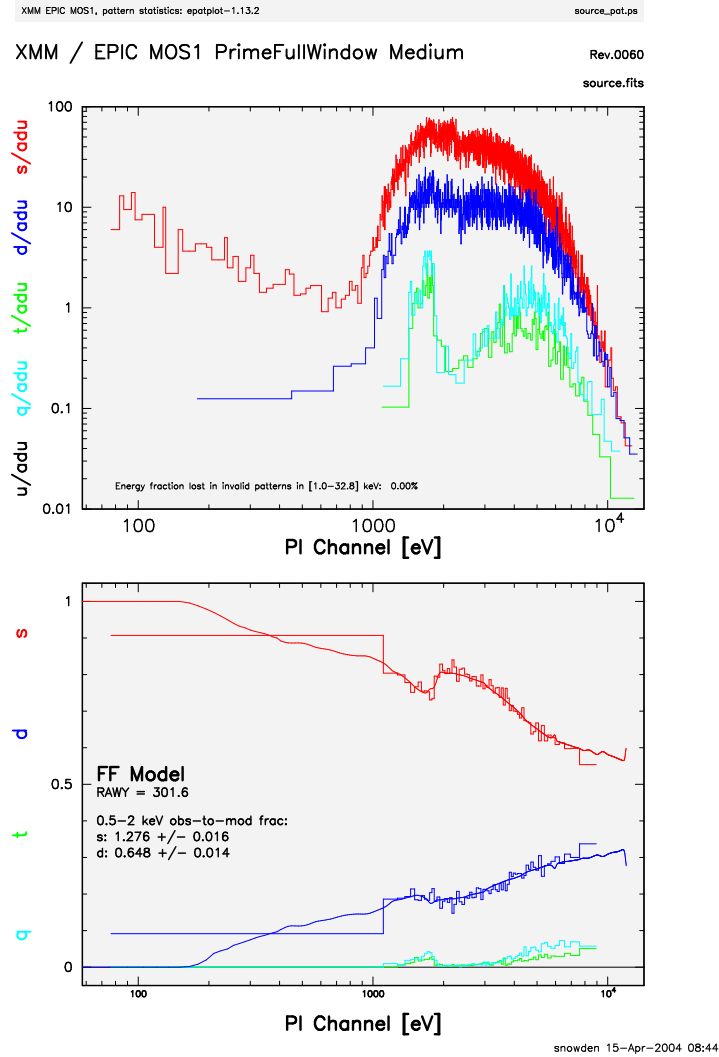


Figure 6.4: The output of *epatplot* for a very faint source without pileup. Note that in the lower plot, for energies less than ~ 1500 eV, there are too few X-rays for *epatplot* to model, and for energies higher than ~ 1500 eV, there is excellent agreement between the expected and observed pattern distributions.

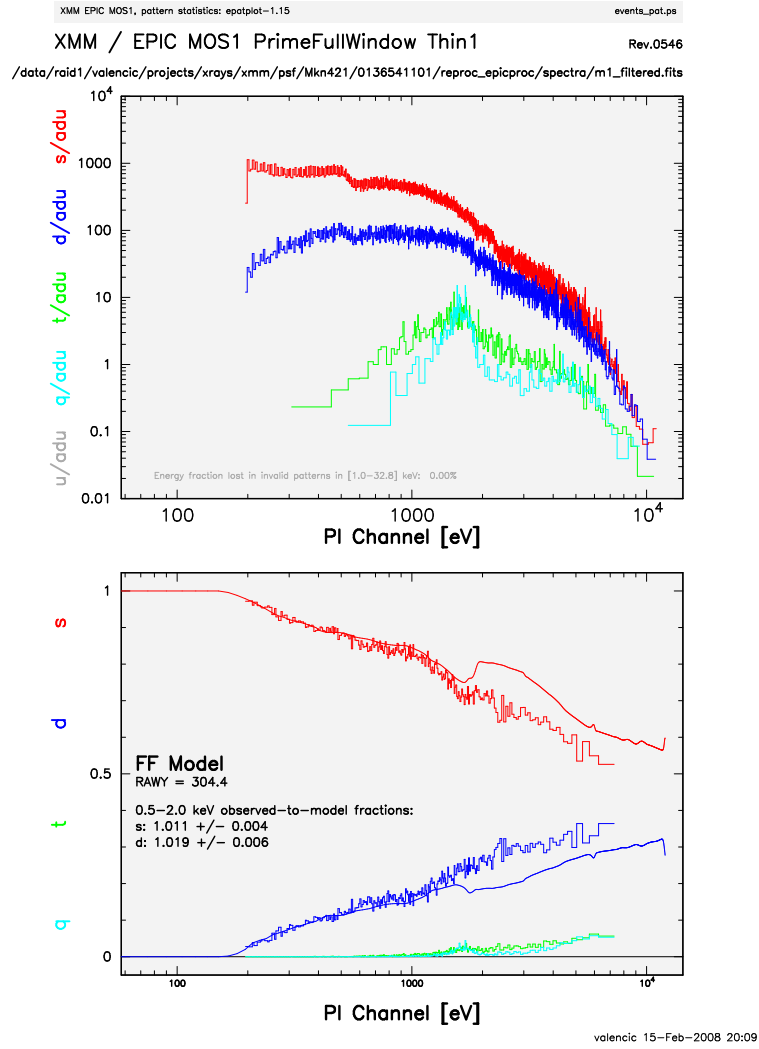


Figure 6.5: The output of *epatplot* for a heavily piled source. In the lower plot, there are large differences between the predicted and observed pattern distribution at energies above ~ 1000 eV.

6.2.10 Create the Photon Redistribution Matrix (RMF) and Ancillary File (ARF)

The following assumes that an appropriate source spectrum, named `mos1_pi.fits`, has been extracted as in §6.2.6.

To make the RMF, type

```
rmfgen rmfset=mos1_rmf.fits spectrumset=mos1_pi.fits
```

where

```
rmfset – output file
spectrumset – spectrum file
```

Now use the RMF, spectrum, and event file to make the ancillary file.

To make the ARF, type

```
arfgen arfset=mos1_arf.fits spectrumset=mos1_pi.fits withrmfset=yes
      rmfset=mos1_rmf.fits withbadpixcorr=yes badpixlocation=mos1_filt_time.fits
```

where

```
arfset – output ARF file name
spectrumset – input spectrum file name
withrmfset – flag to use the RMF
rmfset – RMF file created by rmfgen
withbadpixcorr – flag to include the bad pixel correction
badpixlocation – file containing the bad pixel information; should be set to the event
                  file from which the spectrum was extracted.
```

At this point, the spectrum is ready to be analyzed, so skip ahead to prepare the spectrum for fitting (§6.7).

6.3 Examine and Analyze Data on the Command Line (Timing Mode)

The process for analyzing PN Timing data is largely similar to that for Imaging data, but with a few differences. We will assume that the pipeline was rerun and that the output event file was named `pn.fits`.

6.3.1 Create and Display an Image

To create an image, type

```
evselect table=pn.fits withimageset=yes imageset=image.fits
xcolumn=RAWX ycolumn=RAWY imagebinning=binSize ximagebinsize=1 yimagebinsize=1
```

where

`table` – input event table
`withimageset` – make an image
`imageset` – name of output image
`xcolumn` – event column for X axis
`ycolumn` – event column for Y axis
`imagebinning` – form of binning, force entire image into a given size or bin by a specified number of pixels
`ximagebinsize` – binning factor for X axis
`yimagebinsize` – binning factor for Y axis

The output file `image.fits` can be viewed by using a standard FITS display, such as *ds9* (see Figure 6.6).

```
ds9 image.fits &
```

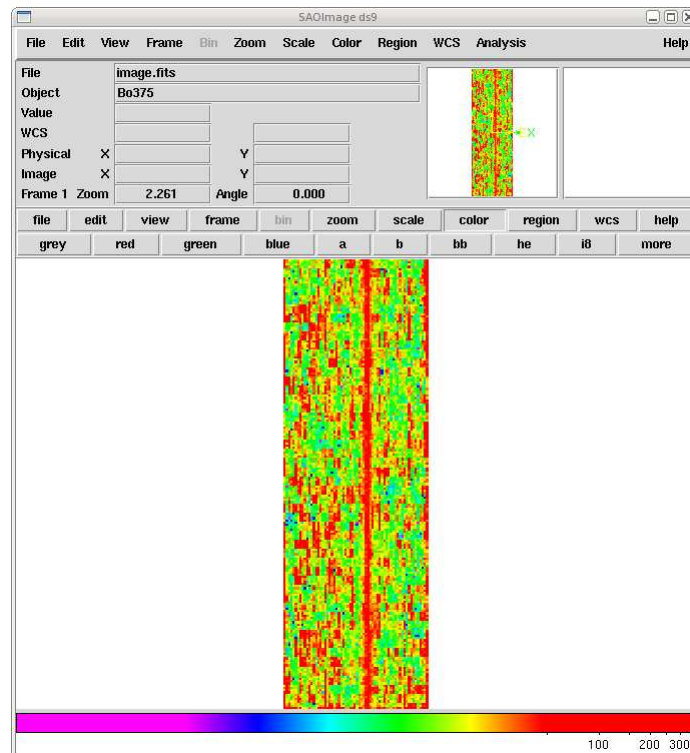


Figure 6.6: An image of PN data taken in Timing mode, displayed in *ds9*.

6.3.2 Create and Display a Light Curve

To create a light curve, type

```
evselect table=pn.fits withrateset=yes rateset=pn_ltrcv.fits
      maketimecolumn=yes timecolumn=TIME timebinsize=50 makeratecolumn=yes
```

where

`table` – input event table
`withrateset` – make a light curve
`rateset` – name of output light curve file
`maketimecolumn` – control to create a time column
`timecolumn` – time column label
`timebinsize` – time binning (seconds)
`makeratecolumn` – control to create a count rate column, otherwise a count column will be created

The output file `pn_ltrcv.fits` can be viewed by using *fv* (see Fig 6.7).

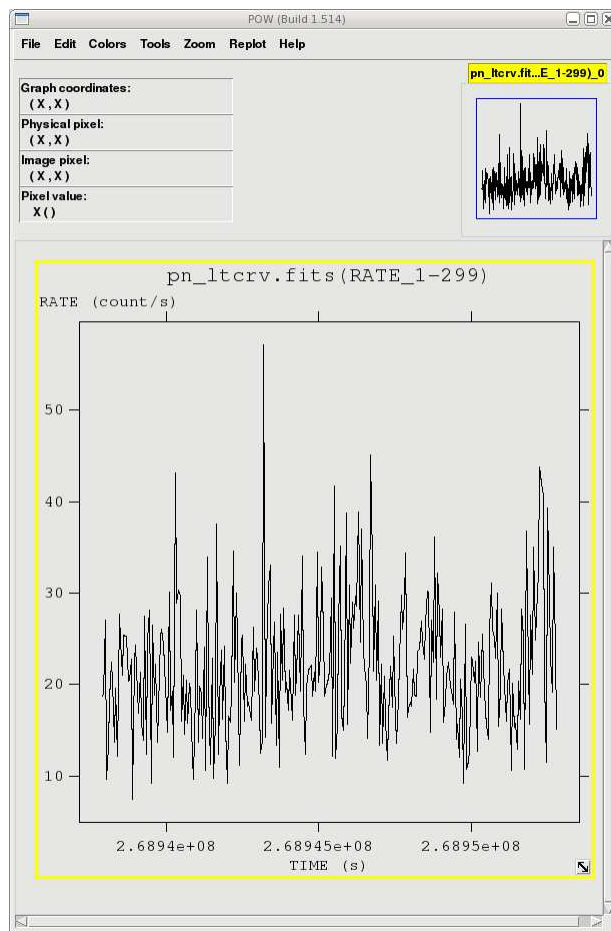


Figure 6.7: The light curve of our example PN Timing mode data, displayed in *fv*.

6.3.3 Applying Standard Filters to the Data

The filtering expressions for PN in Timing Mode is:

```
(PATTERN <= 4)&&(PI in [200:15000])&&#XMMEA_EP
```

The first expression will select good events with `PATTERN` between 0 and 4. The `PATTERN` value is similar the `GRADE` selection for *ASCA* data, and is related to the number and pattern of the CCD pixels triggered for a given event. The `PATTERN` assignments are: single pixel events: `PATTERN == 0`, double pixel events: `PATTERN in [1:4]`, triple and quadruple events: `PATTERN in [5:12]`.

The second keyword in the expression, `PI`, selects the preferred pulse height of the event. Since we are working with PN data, this should be between 200 and 15000 eV. This should clean up the image significantly with most of the rest of the obvious contamination due to low pulse height events. Setting the lower `PI` channel limit somewhat higher (e.g., to 300 eV) will eliminate much of the rest.

Finally, the `#XMMEA_EP` filter provides a canned screening set of `FLAG` values for the event. (The `FLAG` value provides a bit encoding of various event conditions, e.g., near hot pixels or outside of the field of view.) Setting `FLAG == 0` in the selection expression provides the most conservative screening criteria and should always be used when serious spectral analysis is to be done on PN data.

To filter the data, type

```
evselect table=pn.fits withfilteredset=yes
expression='(PATTERN <= 4)&&(PI in [200:15000])&&#XMMEA_EP'
filteredset=pn_filt.fits filtertype=expression keepfilteroutput=yes
updateexposure=yes filterexposure=yes
```

where

```
table – input event table
filtertype – method of filtering
expression – filtering expression
withfilteredset – create a filtered set
filteredset – output file name
keepfilteroutput – save the filtered output
updateexposure – for use with temporal filtering
filterexposure – for use with temporal filtering
```

6.3.4 Applying Time Filters to the Data

Sometimes, soft proton background flaring makes it necessary to use filters on time in addition to those mentioned above.

To determine if there is flaring in the observation, make a light curve and display it, as demonstrated in §6.3.2 and plotted in Figure 6.7. No flares are evident, so we will continue to the next section. However, if a given dataset does contain flaring, it should be removed in the same way as shown in §6.2.4.

6.3.5 Extract the Source and Background Spectra

First, we will need to make an image of the filtered event file following the procedures set out in §6.3.1:

```
evselect table=pn_filt.fits imagebinning=binSize
imageset=image.fits withimageset=yes
xcolumn=RAWX ycolumn=RAWY ximagebinsize=1 yimagebinsize=1
```

where the keywords are as described in §6.3.1. The image can be displayed with `ds9`. As can be seen in Figure 6.6, the source is centered on `RAWX=38`. We will extract this and the 2 pixels on either side of it.

```
evselect table=pn_filt.fits withspectrumset=yes spectrumset=source.pi.fits
energycolumn=PI spectralbinsize=5 withspecranges=yes specchannelmin=0
specchannelmax=20479 expression='(FLAG==0) && (RAWX in [36:40])' withfilteredset=yes
filteredset=pn_filt_source.fits
```

For the background, the extraction area should be as far from the source as possible.

```
evselect table=pn_filt.fits withspectrumset=yes spectrumset=bkg.pi.fits
energycolumn=PI spectralbinsize=5 withspecranges=yes specchannelmin=0
expression='(FLAG==0) && (RAWX in [3:5])'
specchannelmax=20479 withfilteredset=y filteredset=pn_filt_bkg.fits
```

6.3.6 Determine the Spectrum Extraction Areas

The source and background region areas can now be found. (This process is identical to that used for Imaging data.) This is done with the task *backscale*, which takes into account any bad pixels or chip gaps, and writes the result into the BACKSCAL keyword of the spectrum table.

To find the source and background extraction areas, type

```
backscale spectrumset=source_pi.fits badpixlocation=pn_filt.fits
```

and

```
backscale spectrumset=bkg_pi.fits badpixlocation=pn_filt.fits
```

6.3.7 Check for Pile Up

Depending on how bright the source is and what modes the EPIC detectors are in, event pile up may be a problem. Pile up occurs when a source is so bright that incoming X-rays strike two neighboring pixels or the same pixel in the CCD more than once in a read-out cycle. In such cases the energies of the two events are in effect added together to form one event. Pile up and how to deal with it are discussed at length in §6.2.8 and §6.2.9, respectively, and users are strongly encouraged to refer to those sections. Briefly, we deal with it in PN Timing data essentially the same way as in Imaging data, that is, by using only single pixel events, and/or removing the regions with very high count rates, checking the amount of pile up, and repeating until it is no longer a problem.

In our example case, the source is far too faint to be piled up, but it is always a good idea to check anyway. To check for pile up, type

```
epatplot set=pn_filt_source.fits plotfile=pn_epat.ps useplotfile=yes
withbackgroundset=yes backgroundset=pn_filt_bkg.fits
```

The output is shown in Fig. 6.8.

6.3.8 My Timing Mode Data is Piled Up! Now What?

There are a couple ways to deal with pile up. First, you can use event file filtering procedures to include only single pixel events (*PATTERN==0*), as these events are less sensitive to pile up than other patterns.

You can also excise areas of high count rates, i.e., the boresight column and several columns to either side of it. (This is analogous to removing the inner-most regions of a source in Imaging data.) The spectrum can then be re-extracted and you can continue your analysis on the excised event file. As with Imaging data, it is recommended that you take an iterative approach: remove an inner region, extract a spectrum, check with *epatplot*, and repeat, each time removing a slightly larger region, until the model and observed pattern distributions agree.

For example, a bright source is shown in Fig. 6.9. We can extract the boresight column and several columns to either side:

```
evselect table=pn_filt.fits withspectrumset=yes spectrumset=source_pi_WithBore.fits
energycolumn=PI spectralbinsize=5 withspecranges=yes specchannelmin=0
specchannelmax=20479 expression='(FLAG ==0) && (RAWX in [28:48])'
withfilteredset=yes filteredset=pn_filt_source_WithBore.fits
```

And just the columns to either side of the boresight, with lower count rates:

```
evselect table=pn_filt.fits withspectrumset=yes spectrumset=source_pi_NoBore.fits
energycolumn=PI spectralbinsize=5 withspecranges=yes specchannelmin=0
expression='(FLAG ==0) && (RAWX in [28:48] && (RAWX in [34:41])'
specchannelmax=20479 withfilteredset=yes filteredset=pn_filt_source_NoBore.fits
```

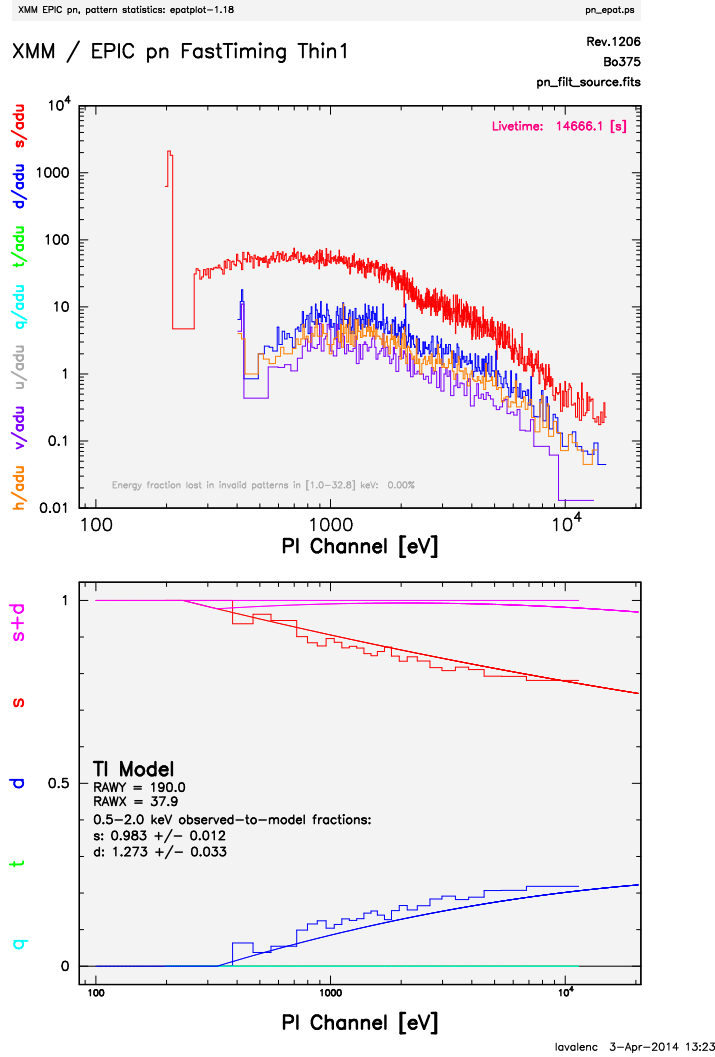



Figure 6.8: The output of *epatplot*.

and check with *epatplot* as we go along until no pile up remains.

Be aware that if you do this, you will need to use a non-standard way to make the ancillary files (ARFs) for your spectrum! This is discussed further in §6.3.9. You will need the spectra of the full extraction area and the excised area, so we might as well get them now. We already have it for the full extraction area, so for the excised area,

```
evselect table=pn_filt.fits withspectrumset=yes spectrumset=source_pi_Excised.fits
energycolumn=PI spectralbinsize=5 withspecranges=yes specchannelmin=0
specchannelmax=20479 expression='RAWX in [34:41]'
withfilteredset=yes filteredset=pn_filt_source_Excised.fits
```

6.3.9 Create the Photon Redistribution Matrix (RMF) and Ancillary File (ARF)

Making the RMF for PN data in Timing mode is exactly the same as in Imaging mode, which is demonstrated in §6.2.10. If the boresight was not excised, making the ARF is the same, as well.

To make the RMF,

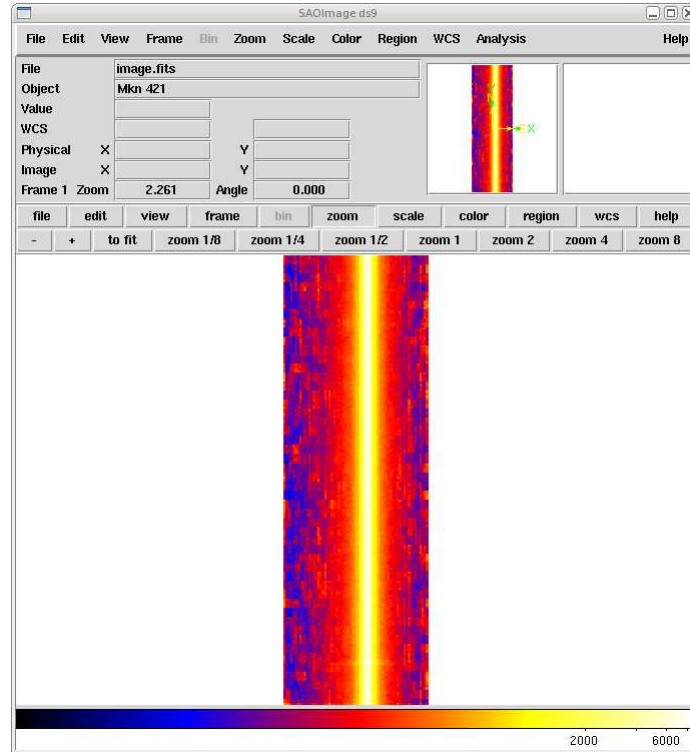


Figure 6.9: An image of a bright source in PN Timing mode.

```
rmfgen rmfset=source_rmf.fits spectrumset=source_pi.fits
```

where

```
rmfset – output file
spectrumset – spectrum file
```

and to make the ARF,

```
arfgen spectrumset=source_pi.fits arfset=source_arf.fits withrmfset=yes
      rmfset=source_rmf.fits badpixlocation=pn_filt.fits detmaptype=psf
```

However, if you had to remove piled up sections, you will need to make an ARF for the full extraction area, another one for the piled up area, and then subtract the two to find the ARF for the non-piled regions. We already have spectra for the full extraction area and the excised area, so we will use them to make the ARFs:

```
arfgen arfset=source_arf_WithBore.fits spectrumset=source_pi_WithBore.fits detmaptype=psf
```

```
arfgen arfset=source_arf_Excised.fits spectrumset=source_pi_Excised.fits detmaptype=psf
```

where

```
arfset – output ARF file name
spectrumset – input spectrum file name
```

Now we can subtract them:

```
addarf "source_arf_WithBore.fits source_arf_Excised.fits" "1.0 -1.0" source_arf.fits
```

At this point, the spectrum is ready to be analyzed, so skip ahead to prepare the spectrum for fitting (§6.7).

6.4 Rerun the Pipeline with the GUI

We assume that the data was prepared and environment variables were set according to §5. In the window where SAS was initialized, in your “processing directory” PROC, run *emchain* or *emproc* (to produce calibrated photon event files for the MOS cameras), and *epchain* or *epproc* (to do the same for the PN camera). If you are reprocessing PN data in TIMING or BURST modes and using *epchain*, you will need to set the relevant parameter, as shown below.

In the window where SAS was initialized, in your “processing directory” PROC, invoke the SAS GUI (see Figure 6.10).

On the command line, type

```
sas &
```

A window will appear with two panels. The upper one will contain task names, what group they belong to (such as utility, epic, timing, calibration, etc.) and a short description of each. The lower one will contain information about environment variables, and as tasks are invoked, feedback from the tasks.

From the upper window, select *emchain* or *emproc* to process the MOS data, and *epchain* or *epproc* to process the PN data. Double-clicking the task will bring up pop-up windows that will allow you to change the (many) parameters; however, for most cases, the default settings are fine, so just click “Run”.

If the dataset has more than one exposure, a specific exposure can be accessed with *epchain* by setting the *exposure* parameter in the “General” tab to the exposure number.

To create an out-of-time event file for your PN data, toggle the parameter *withoutoftime* on the *Epevents* tab to “yes”.

If you have PN data taken in Timing or Burst mode, note that *epchain* assumes by default that the data was taken in Imaging mode. To repipeline PN Timing or Burst data, toggle the *datamode* parameter in the *General* tab to the appropriate value.

By default, none of these tasks keep any intermediate files they generate. *Epchain* maintains the naming convention described in §3.3.3. *Emproc* and *epproc* designate their output event files with “Evts.ds”. “*ImagingEvts.ds”, “*TimingEvts.ds”, and “*BurstEvts.ds” denote the imaging mode, timing, and burst mode event lists, respectively. In either case, you may want to name the new files something easy to type. For example, to rename one of the new MOS1 event files output from *emchain* or *emproc*, respectively, type

```
mv P0123700101M1S001MIEVLI0000.FIT mos1.fits
```

or

```
mv 0070_0123700101_EMOS1_S001_ImagingEvts.ds mos1.fits
```

Once the new event files have been obtained, the analysis techniques described below can be used for data in Imaging Mode. For data in Timing Mode, see §6.6.

```
mv 0070_0123700101_EMOS1_S001_ImagingEvts.ds mos1.fits
```

Once the new event files have been obtained, the analysis techniques described below can be used.

6.5 Examine and Analyze Data with the GUI (Imaging Mode)

Remember that tasks place output files in whatever directory you happened to be in when the SAS GUI was called, so either open and close the GUI in the directory where you want the output or move the files to the directory they should be in.

6.5.1 An Introduction to *xmmselect*

The task *xmmselect* is used for many procedures in the GUI. Like all tasks, it can easily be invoked by starting to type the name and pressing enter when it is highlighted.

When *xmmselect* is invoked a dialog box will first appear requesting a file name. You can either use the browser button or just type the file name in the entry area, “mos1.fits:EVENTS” in this case. To use the

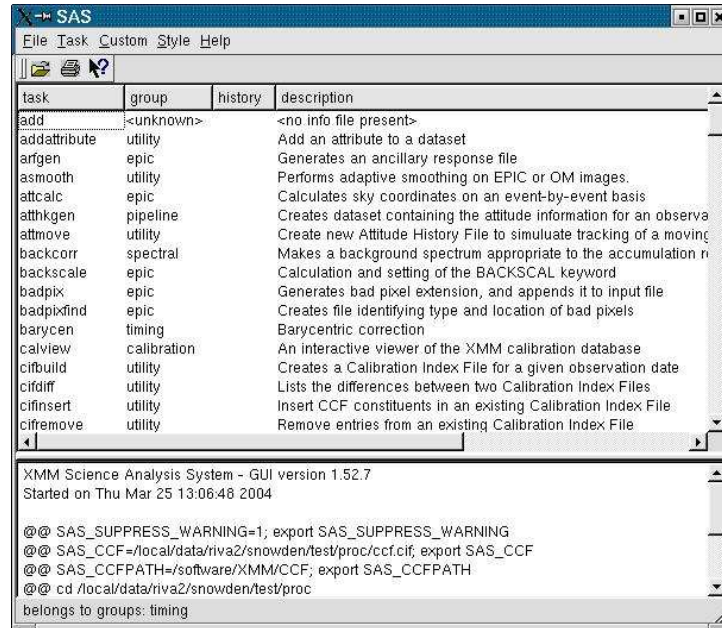


Figure 6.10: The SAS GUI. To locate and invoke a task one need only start typing the task name, and when it is highlighted, press carriage return. Otherwise, double-click on the task name.

browser, select the file folder icon; this will bring up a second window for the file selection. Choose the desired event file, then the “EVENTS” extension in the right-hand column, and click “OK”. The directory window will then disappear and you can click “Run” on the selection window.

When the file name has been submitted the *xmmselect* GUI (see Figure 6.11) will appear, along with a dialog box offering to display the selection expression. The selection expression will include the filtering done to this point on the event file, which for the pipeline processing includes for the most part CCD and GTI selections.

6.5.2 Create and Display an Image

To create an image in sky coordinates by using the *xmmselect*, call *xmmselect* and load the event file as in §6.5.1. Then,

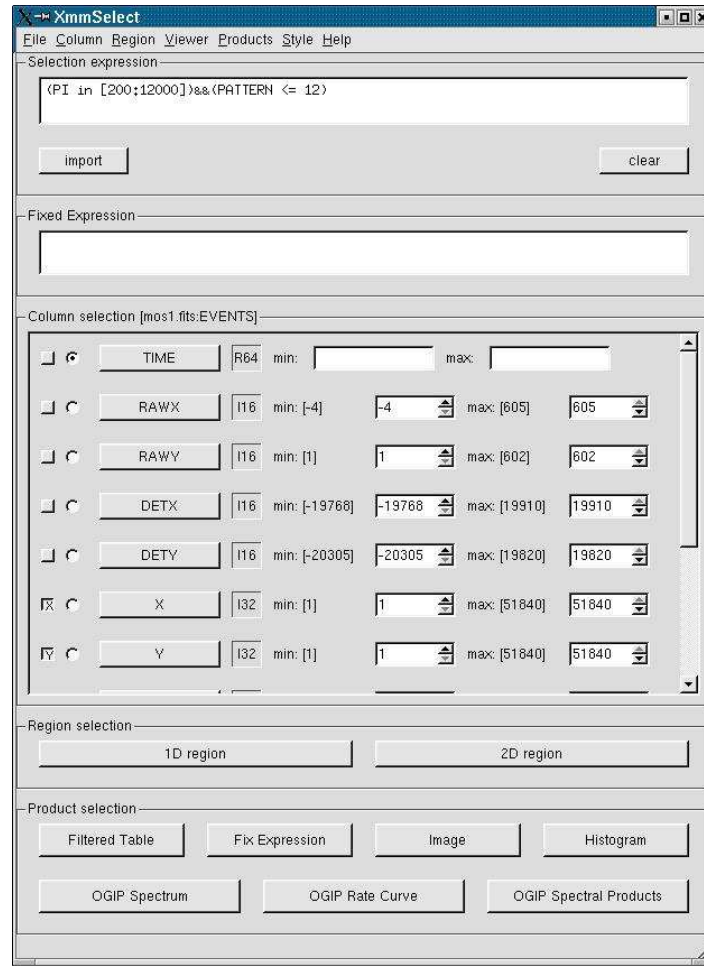
- 1) Check the square boxes to the left of the “X” and “Y” entries.
- 2) Click the “Image” button near the bottom of the page. This brings up the *evselect* GUI (see Figure 6.12).
- 3) In the **imageset** box, enter the name of the output file, in this case, **image.fits**.
- 4) Click the “Run” button on the lower left corner of the *evselect* GUI.

Different binnings and other selections can be invoked by accessing the “Image” tab at the top of the GUI. The default settings are reasonable, however, for a basic image. The resultant image is written to the file **image.fits**, and is automatically displayed using *ds9* (see Figure 6.1).

6.5.3 Create and Display a Light Curve

To create a light curve with *xmmselect*,

- 1) Check the round box to the left of the “Time” entry.
- 2) Click on the “OGIP Rate Curve” button near the bottom of the page. This brings up the *evselect* GUI (see Figure 6.12).
- 3) Click on the “Lightcurve” tab and change the “timebinsize” to a reasonable amount, e.g. 10 or 100 s. In the “rateset” textbox, enter the name of the output file, for example, **mos1_ltrcv.fits**.

Figure 6.11: The *xmmselect* GUI.

- 4) Click on the “Run” button at the lower left corner of the *evselect* GUI.

The resultant light curve is displayed automatically using Grace (see Figure 6.2). It can also be viewed using *fv* or *dsplot*; on the command line, type

```
dsplot table=mos1_ltcv.fits x=TIME y=RATE.ERROR withoffsetx=yes &
```

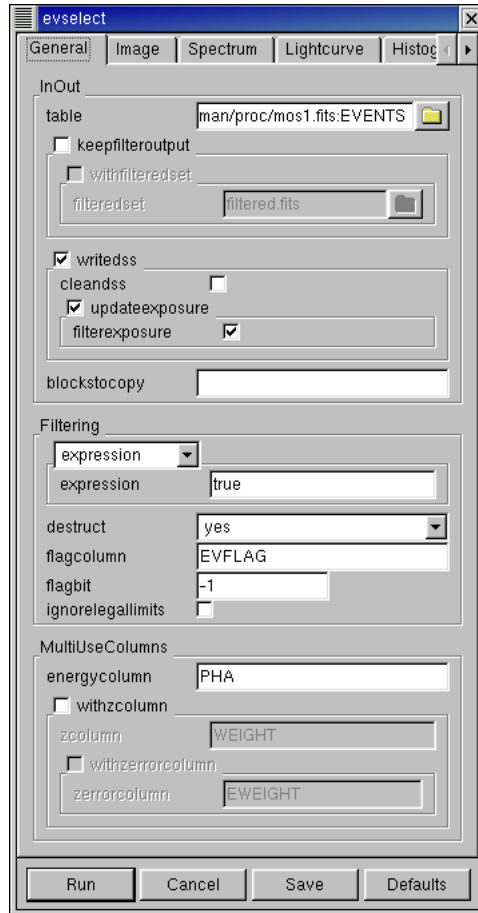
where

```
table – input event table
x – column for plotting on X axis
y – column for plotting on Y axis
withoffsetx – creates an offset to the X axis
```

or, alternatively,

```
fv mos1_ltcv.fits &
```

In the *fv* pop-up window, the RATE extension will be available in the second row (index 1, as numbering begins with 0). Select “PLOT” from this row, and select the column name and axis on which to plot it.

Figure 6.12: The *evselect* GUI.

6.5.4 Applying Standard Filters the Data

The filtering expressions for the MOS and PN are:

```
(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM
```

and

```
(PATTERN <= 12)&&(PI in [200:15000])&&#XMMEA_EP
```

The first two expressions will select good events with **PATTERN** in the 0 to 12 range. The **PATTERN** value is similar the **GRADE** selection for *ASCA* data, and is related to the number and pattern of the CCD pixels triggered for a given event. The **PATTERN** assignments are: single pixel events: **PATTERN** == 0, double pixel events: **PATTERN** in [1:4], triple and quadruple events: **PATTERN** in [5:12].

The second keyword in the expressions, **PI**, selects the preferred pulse height of the event; for the MOS, this should be between 200 and 12000 eV. For the PN, this should be between 200 and 15000 eV. This should clean up the image significantly with most of the rest of the obvious contamination due to low pulse height events. Setting the lower **PI** channel limit somewhat higher (e.g., to 300 eV) will eliminate much of the rest.

Finally, the **#XMMEA_EM** (**#XMMEA_EP** for the PN) filter provides a canned screening set of **FLAG** values for the event. (The **FLAG** value provides a bit encoding of various event conditions, e.g., near hot pixels or outside of the field of view.) Setting **FLAG** == 0 in the selection expression provides the most conservative screening criteria and should always be used when serious spectral analysis is to be done on the PN. It typically is not necessary for the MOS.

It is a good idea to keep the output filtered event files and use them in your analyses, as opposed to re-filtering the original file with every task. This will save much time and computer memory. As an example, the

Lockman Hole data’s original event file is 48.4 Mb; the fully filtered list (that is, filtered spatially, temporally, and spectrally) is only 4.0Mb!

To filter the data using *xmmselect*,

- 1) Enter the filtering criteria in the “Selection Expression” area at the top of the *xmmselect* window:

```
(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM
```

- 2) Click on the “Filtered Table” box at the lower left of the *xmmselect* GUI.
- 3) Change the *evselect filteredset* parameter, the output file name, to something useful, e.g., *mos1_filt.fits*
- 4) Click “Run”.

6.5.5 Applying Time Filters the Data

Sometimes, it is necessary to use filters on time in addition to those mentioned above. This is because of soft proton background flaring, which can have count rates of 100 counts/sec or higher. It should be noted that the amount of flaring that needs to be removed depends in part on the object observed; a faint, extended object will be more affected than a very bright X-ray source.

There are two ways to filter on time: with an explicit reference to the **TIME** or **RATE** parameters in the filtering expression, or by creating a secondary Good Time Interval (GTI) file with the task *tabgtigen*. Both procedures are described below. For the example data, we will filter by time, though you can just as easily filter by rate.

To explicitly define the **TIME** or **RATE** parameters,

- 1) Make a light curve and display it in Grace (see §6.5.3).
- 2) In the Grace window, pull down the “Edit” menu, select “Regions”, and select “Define”; for the Lockman case, select “Left of Line” for the “Region type”.
- 3) Click the “Define” button and then click at two points to create a vertical line at the upper end of the desired range on the Grace plot. (It is possible to define up to five regions at one time by changing the “Define region” counter.)
- 4) Back on the *xmmselect* GUI, click on the “1D region” button. This will transfer the selection criteria to the “Selection expression” location.

There is a very large flare toward the end of the observation, so the syntax for the time selection is **(TIME <= 73227600)**. However, there is also a small flare within an otherwise good interval. A slightly more complicated expression to remove it would be: **(TIME <= 73227600)&&!(TIME IN [73221920:73223800])**. The syntax **&&(TIME < 73227600)** includes only events with times less than 73227600, and the “!” symbol stands for the logical “not”, so use **&&!(TIME in [73221920:73223800])** to exclude events in the time interval 73221920 to 73223800.

If combined with the standard filtering expression (see §6.5.4), the full filtering expression would then be:

```
(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM
&&(TIME <= 73227600) &&!(TIME in [73221920:73223800])
```

This expression can then be used to filter the original event file, as shown in §6.5.4, or only the times can be used to filter the file that has already had the standard filters applied:

- 1) Load the filtered file “*mos1_evt2.fits*” in *xmmselect*.
- 2) Enter the filtering criteria in the “Selection Expression” area at the top of the *xmmselect* GUI:

```
(TIME <= 73227600) &&!(TIME in [73221920:73223800])
```

- 3) Click on the “Filtered Table” box at the lower left of the *xmmselect* GUI.

- 4) Change the *evselect* **filteredset** parameter, the output file name, to something useful, e.g., `mos1_filt.fits`
- 5) Click “Run”.

To filter on time using a secondary GTI file, make the GTI by using the same time filtering parameters as determined above and the *tabgtigen* task:

- 1) Call up *tabgtigen* in the SAS window and select the filtered event file, `mos1_filt.fits`, and the extension, **EVENTS**.
- 2) Edit the output file name, the **gtiset** parameter; here, we will use `gtiset.fits`.
- 3) Enter the filtering expression. In this case, it is `(TIME <= 73227600) &&! (TIME in [73221920:73223800])`
- 4) Click “Run”.

The new GTI file can be applied with *xmmselect*:

- 1) Invoke *xmmselect* and load the event file, `mos1_filt.fits`.
- 2) In the “Selection Expression” box, type `GTI(gtiset.fits,TIME)`,
- 3) Click on the “Filtered Table” box at the lower left of the *xmmselect* GUI.
- 4) Change the *evselect* **filteredset** parameter, the output file name, to something useful; here, we will use `mos1_filt_time.fits`.

6.5.6 Source Detection with *edetect_chain*

The *edetect_chain* task does nearly all the work involved with EPIC source detection, but it requires some input files that must be generated first. The first step is to make an attitude file with the task *atthkgen*, then images of the sources in the desired energy bands, with the image binning sizes as needed according to the detector. For the MOS, the binsize is 22; for the PN, the binsize is 82.

In the example below, source detection is done only on MOS1 images in two bands (“soft” X-rays with energies between 300 and 2000 eV, and “hard” X-rays, with energies between 2000 and 10000 eV).

The example uses the filtered event file produced in §6.5.5, with the assumption that it is located in the current directory.

First, make the attitude file.

- 1) Call the task *atthkgen*.
- 2) Set the **timestep** keyword to 1. Set the **atthkset** keyword to the desired output file name, for example, `attitude.fits`.
- 3) Click “Run”.

Next, make the soft and hard X-ray images.

- 1) Call the task *evselect*.
- 2) In the “General” tab, set the “Table” parameter to the MOS1 event file name (`mos1_filt_time.fits`). Confirm that “Filtertype” is set to “expression”, and in the “Expression” text area, type `(FLAG == 0)&&(PI in [300:2000])`. In the “Image” tab, check the **withimageset** box, and enter the desired output image name, for example, `mos1-s.fits`. Set **xcolumn** to X and **ycolumn** to Y. Set **Binning** to **binSize**, **ximagebinsize** to 22, and **yimagebinsize** to 22. For the display purposes, we will also make an image over both bands, which we will call `mos1-all.fits`.
- 3) Click “Run”.
- 4) Call the task *evselect* again.

- 5) In the “General” tab, set the “Table” parameter to the MOS1 event file name (`mos1_filt_time.fits`). Confirm that “Filtertype” is set to “expression”, and in the “Expression” text area, type `(FLAG == 0)&&(PI in [2000:10000])`. In the “Image” tab, check the `withimageset` box, and enter the desired output image name, for example, `mos1-h.fits`. Confirm that `xcolumn` is set to X, `ycolumn` is set to Y, Binning is `binSize`, and `ximagebinsize` and `yimagebinsize` are both 22.
- 6) Click “Run”.

Now we can run `edetect_chain`.

- 1) Call the task `edetect_chain`.
- 2) In the “0” tab, in the `imagesets` text area, type: `mos1-s.fits mos1-h.fits` In the `eventsets` area, enter the names of the event file: `mos1_filt_time.fits`. In the `attitudeset` area, enter the name of the attitude file made by the task `atthkgen` (`attitude.fits`). Set the `pimin` keyword to the minimum PI values (in eV) for the input images by typing: 300 2000, and do similar for the maximum values for `pimax` (2000 10000). Set the `likemin` parameter to 10, `withexpmap` to `yes`, `ecf` to 0.878 0.220. In the “1” tab, set `eboxl_list` to `eboxlist_l.fits` and `eboxm_list` to `eboxlist_m.fits`. In the “2” tab, set `esp_withootset` to `no` and `eml_list` to `emllist.fits`.
- 3) Click “Run”.

The energy conversion factors (ECFs) convert the source count rates into fluxes. The ECFs for each detector and energy band depend on the pattern selection and filter used during the observation. For more information, please consult the calibration paper “SSC-LUX-TN-0059”, available at the XMM-Newton Science Operations Center or see Table 8 in the 3XMM Catalogue User Guide. Those used here are derived from PIMMS using the flux in the 0.1-10.0 keV band, a source power-law index of 1.9, an absorption of 0.5×10^{20} .

We can display the results of `eboxdetect` using the task `srcdisplay` and produce a region file for the sources.

- 1) Call the task `srcdisplay`.
- 2) Set `boxlistset` to `emllist.fits`. Confirm that `withimageset` is checked, and set `imageset` to `mos1-all.fits`. Check `withregionfile`, and set `regionfile` to `regionfile.txt`. Confirm that `sourceradius` is set to 0.01.
- 3) Click “Run”.

Figure 6.3 shows the MOS1 image overlayed with the detected sources.

6.5.7 Extract the Source and Background Spectra

Throughout the following, please keep in mind that some parameters are instrument-dependent. The parameter `specchannelmax` should be set to 11999 for the MOS, or 20479 for the PN. Also, for the PN, the most stringent filters, `(FLAG==0)&&(PATTERN<=4)`, must be included in the `expression` to get a high-quality spectrum.

For the MOS, the standard filters should be appropriate for many cases, though there are some instances where tightening the selection requirements might be needed. For example, if obtaining the best-possible spectral resolution is critical to your work, and the corresponding loss of counts is not important, only the single pixel events should be selected (`PATTERN==0`). If your observation is of a bright source, you again might want to select only the single pixel events to mitigate pile up (see §6.5.9 and §6.5.10 for a more detailed discussion).

In any case, you’ll need to know spatial information about the area over which you want to extract the spectrum, so display the filtered event file with `ds9`:

```
ds9 mos1_filt_time.fits &
```

Click on the object whose spectrum you wish to extract. This will produce a circle (extraction region), centered on the object. The circle’s radius can be changed by clicking on it and dragging to the desired size. Adjust the size and position of the circle until you are satisfied with the extraction region; then, double-click on the region to bring up a window showing the center coordinates and radius of the circle. For this example, we will choose the source at (26188.5,22816.5) and set the extraction radius to 300.

To extract the source spectrum,

- 1) If the filtered file (`mos1_filt_time.fits`) isn't already loaded, invoke *xmmselect* and load it.
- 2) Make an image (see §6.5.2). It will be displayed automatically in a *ds9* window.
- 3) Click on the object whose spectrum you wish to extract. This will produce a circle (extraction region), centered on the object. The circle's radius can be changed by clicking on it. Adjust the size and position of the circle until you are satisfied with the extraction region.
- 4) Click on "2D Region" in the *xmmselect* GUI. This transfers the region information into the "Selection Expression" text area, for example, `((X,Y) IN circle(26188.5,22816,300))`. The `circle` parameters are the X, Y coordinates of the center, and the radius in units of 0.05 arcsec; so, the region used here has a radius of 30 arcsec.
- 5) Click the round button next the PI column on the *xmmselect* GUI.
- 6) Click on "OGIP Spectrum".
- 7) In the "General" page, check `keepfilteroutput` and `withfilteredset`. In the `filteredset` box, enter the name of the event file output, in this case, `mos1_filtered.fits`.
- 8) Select the "Spectrum" page of the *evselect* GUI to set the file name and binning parameters for the spectrum. Confirm that `withspectrumset` is checked. Set `spectrumset` to the desired output name, in this case, `mos1_pi.fits`. Confirm that `withspecranges` is checked. Set `specchannelmin` to 0. Set `specchannelmax` to 11999 for the MOS, or 20479 for the PN.
- 9) Click "Run".

When extracting the background spectrum, follow the same procedures, but change the extraction area. For example, make an annulus around the source; this can be done using two circles, each defining the inner and outer edges of the annulus, then change the filtering expression (and output file name) as necessary.

To extract the background spectrum,

- 1) Follow steps 1 and 2 for extracting the source spectrum.
- 2) Make an annulus around the object whose background spectrum you wish to extract. This can be done using two circles, each defining the inner and outer edges of the annulus.
- 3) Select the inner circle. From "Regions" on the *ds9* pulldown menu, select "Properties", then "Exclude". Adjust the sizes and positions of the circles until you are satisfied with the extraction region.
- 4) In the *xmmselect* GUI, make sure the "Selection Expression" box is empty.
- 5) Click on "2D Region" in the *xmmselect* GUI. Confirm that the correct values were transferred to the "Selection Expression" box; in this case, it should show


```
((X,Y) in CIRCLE(26188.5,22816,1500))&&!((X,Y) in CIRCLE(26188.5,22816,500))
```
- 6) Follow steps 5 - 9 for extracting the source spectrum, giving the output files appropriate names. In this case, let the spectrum be `bkg_pi.fits` and the filtered file be `bkg_filtered.fits`.

6.5.8 Determine the Spectrum Extraction Areas

The source and background region areas can now be found. This is done with the task *backscale*, which takes into account any bad pixels or chip gaps, and writes the result into the `BACKSCAL` keyword of the spectrum table.

To find the source extraction area,

- 1) Invoke *backscale* from the SAS GUI.

- 2) In the “Main” tab, enter the name of the spectrum, `mos1_pi.fits`. In the “Effects” tab, confirm that `withbadpixcorr` is checked, and enter the name of the event file in `badpixlocation`.
- 3) Click “Run”.

Follow the same steps to find the background spectrum area, changing the input spectrum file.

6.5.9 Check for Pile Up

Depending on how bright the source is and what modes the EPIC detectors are in, event pile up may be a problem. Pile up occurs when a source is so bright that incoming X-rays strike two neighboring pixels or the same pixel in the CCD more than once in a read-out cycle. In such cases the energies of the two events are in effect added together to form one event. If this happens sufficiently often, 1) the spectrum will appear to be harder than it actually is, and 2) the count rate will be underestimated, since multiple events will be undercounted. To check whether pile up may be a problem, use the SAS task *epatplot*. (Heavily piled sources will be immediately obvious, as they will have a “hole” in the center.) Note that this procedure requires as input the event files created when the spectrum was made.

The output of *epatplot* is a postscript file, which may be viewed with viewers such as *gv*, containing two graphs describing the distribution of counts as a function of PI channel; see Figure 6.13.

A few words about interpreting the plots are in order. The top is the distribution of counts versus PI channel for each pattern class (single, double, triple, quadruple), and the bottom is the expected pattern distribution (**smooth lines**) plotted over the observed distribution (**histogram**). If the lower plot shows the model distributions for single and double events diverging significantly from the observed distributions, then the source is piled up.

The source used in our Lockman Hole example is too faint to provide reasonable statistics for *epatplot* and is far from being affected by pile up. In contrast, Figure 6.14 shows an example of a bright source (from a different observation) which is strongly affected by pileup. Note the severe divergence between the model and the observed pattern distribution.

To check for pile up,

- 1) Invoke *epatplot* in the SAS GUI.
- 2) In the *epatplot* window, in the “set” text area, enter the name of the event file that was made when the spectrum was extracted, in this case, `mos1_filtered.fits`. Click the “withbackgroundset” box. In the “backgroundset” text area, enter the name of the background event file output when the background spectrum was extracted, in this case, `bkg_filtered.fits`. If you wish to change the output file name to something other than the default, click the “useplotfile” box and enter the name in the “plotfile” text area; For this example, we will use `mos1_epat.ps`.
- 3) Click “Run”.

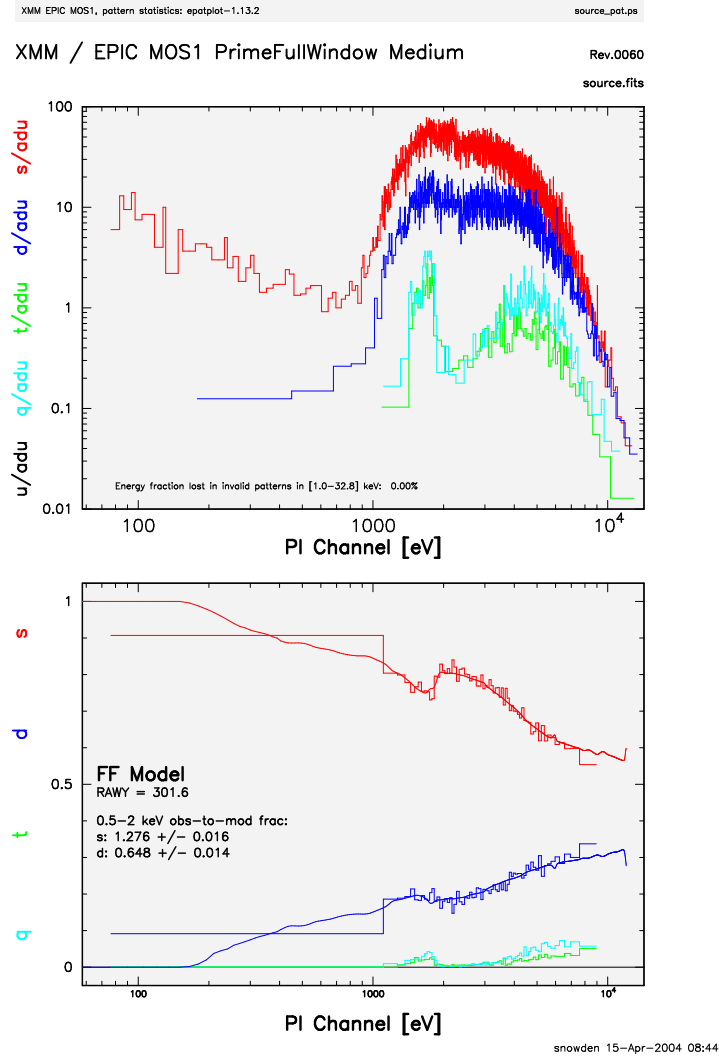


Figure 6.13: The output of *epatplot* for a very faint source without pileup. Note that in the lower plot, for energies less than ~ 1500 eV, there are too few X-rays for *epatplot* to model, and for energies higher than ~ 1500 eV, there is excellent agreement between the expected and observed pattern distributions.

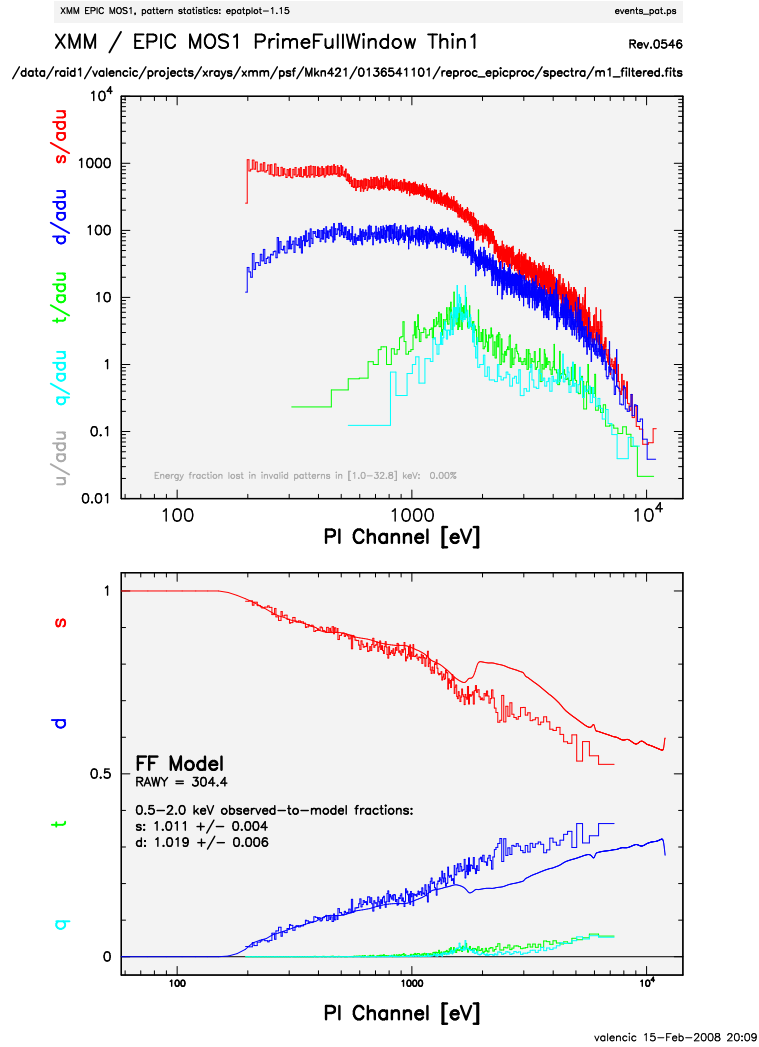


Figure 6.14: The output of *epatplot* for a heavily piled source. In the lower plot, there are large differences between the predicted and observed pattern distribution at energies above ~ 1000 eV.

6.5.10 My Observation is Piled Up! Now What?

There are a few ways to deal with pile up. First, using the region selection and event file filtering procedures demonstrated in earlier sections, you can excise the inner-most regions of a source (as they are the most heavily piled up), re-extract the spectrum, and continue your analysis on the excised event file. For this procedure, it is recommended that you take an iterative approach: remove an inner region, extract a spectrum, check with *epatplot*, and repeat, each time removing a slightly larger region, until the model and observed distribution functions agree.

You can also use the event file filtering procedures to include only those events with `PATTERN==0`, as these events are less sensitive to pile up than other patterns.

6.5.11 Create the Photon Redistribution Matrix (RMF) and Ancillary File (ARF)

The following assumes that an appropriate source spectrum, named `mos1_pi.fits`, has been extracted as in §6.5.7.

To make the RMF,

- 1) Invoke the task *rmfgen* in the SAS GUI.
- 2) In the “Main” tab, set the `spectrumset` keyword to the spectrum file name, e.g., `mos1_pi.fits`. Set the `rmfset` keyword to the RMF file name, e.g., `mos1_rmf.fits`.
- 3) Click “Run”.

To make the ARF,

- 1) Invoke the task *arfgen* in the SAS GUI.
- 2) In the “Main” tab, set the `arfset` parameter to the ARF file name, for example, `mos1_arf.fits`. Set the `spectrumset` parameter to the spectrum file name, in this case, `mos1_pi.fits`.
- 3) In the “Effects” tab, confirm that the `withbadpixcorr` box is checked. Set the `badpixlocation` keyword to the event file name from which the spectrum was extracted, in this case, `mos1_filt_time.fits`.
- 4) In the “Calibration” tab, check the `withrmfset` box and set the `rmfset` keyword to the RMF file name, in this case, `mos1_rmf.fits`.
- 5) Click “Run”.

At this point, the spectrum is ready to be analyzed, so skip ahead to prepare the spectrum for fitting §6.7.

6.6 Examine and Analyze Data with the GUI (Timing Mode)

The process for analyzing PN Timing data is largely similar to that for Imaging data, but with a few differences. We will assume that the pipeline was rerun and that the output event file was named “pn.fits”.

Many popular data products, such as images and light curves, are made with the task *xmmselect*. The following sections detail how to make them. For an introduction to *xmmselect* and a discussion on how to load an event file in it, please see §6.5.1.

6.6.1 Create and Display an Image

To create an image in sky coordinates by using the *xmmselect*, call *xmmselect* and load the event file as in §6.5.1. Then,

- 1) Check the square boxes to the left of the “RAWX” and “RAWY” entries.
- 2) Click the “Image” button near the bottom of the page. This brings up the *evselect* GUI (see Figure 6.12).
- 3) In the “General” tab, in the **imageset** box, enter the name of the output file, in this case, **image.fits**.
- 4) In the “Image” tab, toggle Binning to binSize, and confirm that **ximagebinsize** and **yimagebinsize** are set to 1.
- 5) Click the “Run” button on the lower left corner of the *evselect* GUI.

The output file **image.fits** can be viewed by using a standard FITS display such as *ds9* (see Fig. 6.15).

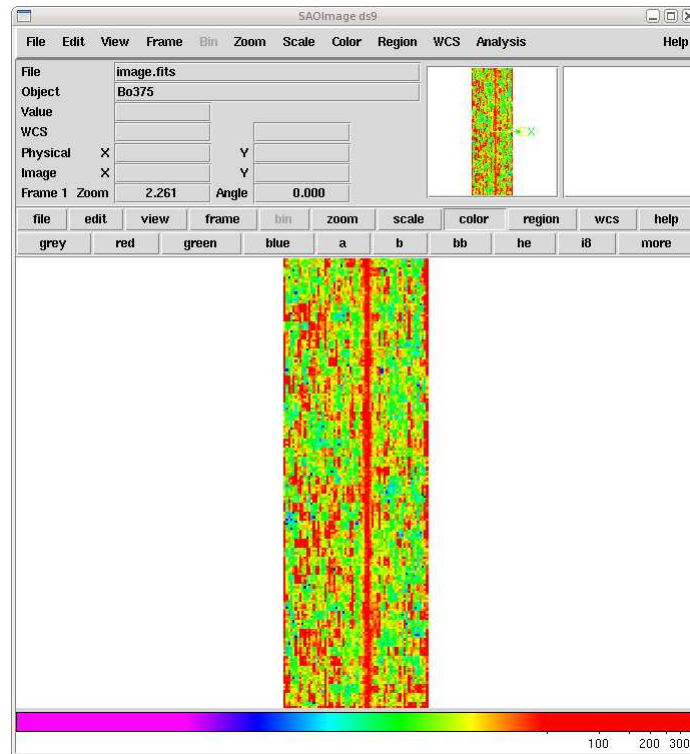


Figure 6.15: An image of PN data taken in Timing mode, displayed in *ds9*.

6.6.2 Create and Display a Light Curve

To create a light curve with *xmmselect*,

- 1) Check the round box to the left of the “Time” entry.
- 2) Click on the “OGIP Rate Curve” button near the bottom of the page. This brings up the *evselect* GUI (see Figure 6.12).
- 3) Click on the “Lightcurve” tab and change the “timebinsize” to a reasonable amount, e.g., 50. In the “rateset” textbox, enter the name of the output file, **pn_ltrcv.fits**.
- 4) Click on the “Run” button at the lower left corner of the *evselect* GUI.

The output file **pn_ltrcv.fits** can be viewed by using *fv* (see Fig 6.16).

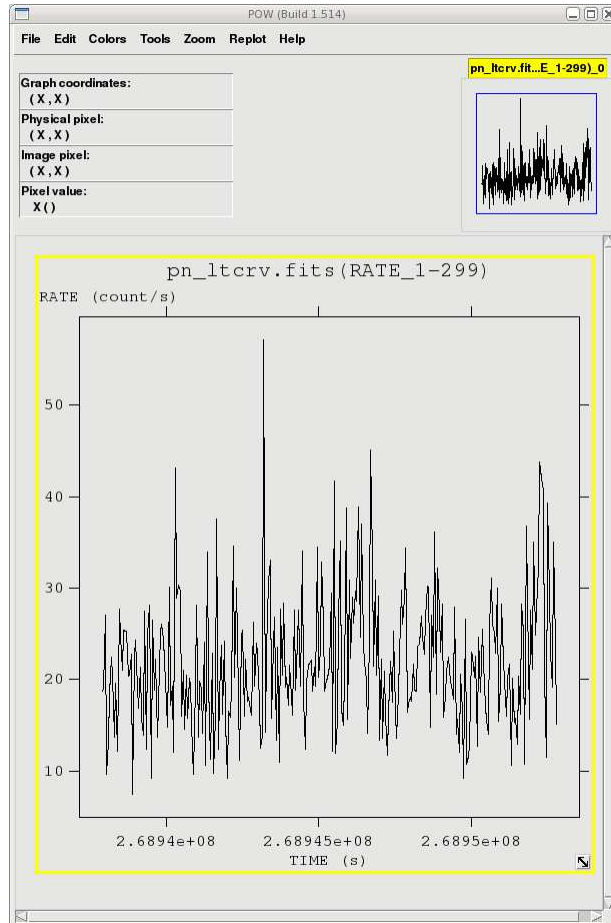


Figure 6.16: The light curve of our example PN Timing mode data, displayed in *fv*.

6.6.3 Applying Standard Filters to the Data

The filtering expressions for PN in Timing Mode is:

```
(PATTERN <= 4)&&(PI in [200:15000])&&#XMMEA_EP
```

The first expression will select good events with **PATTERN** between 0 and 4. The **PATTERN** value is similar to the **GRADE** selection for *ASCA* data, and is related to the number and pattern of the CCD pixels triggered for a given event. The **PATTERN** assignments are: single pixel events: **PATTERN** == 0, double pixel events: **PATTERN** in [1:4], triple and quadruple events: **PATTERN** in [5:12].

The second keyword in the expression, `PI`, selects the preferred pulse height of the event. Since we are working with PN data, this should be between 200 and 15000 eV. This should clean up the image significantly with most of the rest of the obvious contamination due to low pulse height events. Setting the lower `PI` channel limit somewhat higher (e.g., to 300 eV) will eliminate much of the rest.

Finally, the `#XMMEA_EP` filter provides a canned screening set of `FLAG` values for the event. (The `FLAG` value provides a bit encoding of various event conditions, e.g., near hot pixels or outside of the field of view.) Setting `FLAG == 0` in the selection expression provides the most conservative screening criteria and should always be used when serious spectral analysis is to be done on PN data.

To filter the data using *xmmselect*,

- 1) Enter the filtering criteria in the “Selection Expression” area at the top of the *xmmselect* window:

```
(PATTERN <= 4)&&(PI in [200:15000])&&#XMMEA_EP
```

- 2) Click on the “Filtered Table” box at the lower left of the *xmmselect* GUI.
- 3) Change the *evselect* `filteredset` parameter, the output file name, to something useful, e.g., `pn_filt.fits`
- 4) Click “Run”.

6.6.4 Applying Time Filters to the Data

Sometimes, soft proton background flaring makes it necessary to use filters on time in addition to those mentioned above.

To determine if there is flaring in the observation, make a light curve and display it, as demonstrated in §6.6.2 and plotted in Figure 6.16. No flares are evident, so we will continue to the next section. However, if a given dataset does contain flaring, it should be removed in the same way as shown in §6.5.5.

6.6.5 Extract the Source and Background Spectra

First, we will need to make an image of the filtered event file `pn_filt.fits` using *xmmselect*, by following the procedures set out in §6.6.1:

- 1) Check the square boxes to the left of the “RAWX” and “RAWY” entries.
- 2) Click the “Image” button near the bottom of the page. This brings up the *evselect* GUI (see Figure 6.12).
- 3) In the “General” tab, in the `imageset` box, enter the name of the output file, in this case, `image.fits`.
- 4) In the “Image” tab, toggle Binning to `binSize`, and confirm that `ximagebinsize` and `yimagebinsize` are set to 1.
- 5) Click the “Run” button on the lower left corner of the *evselect* GUI.

The image can be displayed with *ds9*. As can be seen in Figure 6.15, the source is centered on `RAWX=38`. We will extract this and the 2 pixels on either side of it.

- 1) Enter the filtering criteria in the “Selection Expression” area at the top of the *xmmselect* window:
(FLAG==0) && (RAWX in [36:40]).
- 2) Click the round button next the `PI` column on the *xmmselect* GUI.
- 3) Click on “OGIP Spectrum”.
- 4) In the “General” tab, check `keepfilteroutput` and `withfilteredset`. In the `filteredset` box, enter the name of the event file output, in this case, `pn_filt_source.fits`.
- 5) In the “Spectrum” tab, set the file name and binning parameters for the spectrum. Confirm that `withspectrumset` is checked. Set `spectrumset` to the desired output name, in this case, `source_pi.fits`. Confirm that `withspecranges` is checked. Set `specchannelmin` to 0 and `specchannelmax` to 20479.

- 6) Click “Run”.

For the background, the extraction area should be as far from the source as possible.

- 1) Enter the filtering criteria in the “Selection Expression” area at the top of the *xmmselect* window:
(FLAG==0) && (RAWX in [3:5]).
- 2) Click the round button next the PI column on the *xmmselect* GUI.
- 3) Click on “OGIP Spectrum”.
- 4) In the “General” tab, check `keepfilteroutput` and `withfilteredset`. In the `filteredset` box, enter the name of the event file output, in this case, `pn_filt_bkg.fits`.
- 5) In the “Spectrum” tab, set the file name and binning parameters for the spectrum. Confirm that `withspectrumset` is checked. Set `spectrumset` to the desired output name, in this case, `bkg_pi.fits`. Confirm that `withspecranges` is checked. Set `specchannelmin` to 0 and `specchannelmax` to 20479.
- 6) Click “Run”.

6.6.6 Determine the Spectrum Extraction Areas

The source and background region areas can now be found. This is done with the task *backscale*, which takes into account any bad pixels or chip gaps, and writes the result into the `BACKSCAL` keyword of the spectrum table.

To find the source extraction area,

- 1) Invoke *backscale* from the SAS GUI.
- 2) In the “Main” tab, enter the name of the spectrum, `source_pi.fits`.
- 3) In the “Effects” tab, confirm that `withbadpixcorr` is checked, and enter the event file, `pn_filt.fits`, in `badpixlocation`.
- 4) Click “Run”.

Follow the same steps to find the background spectrum extraction area.

6.6.7 Check for Pile Up

Depending on how bright the source is and what modes the EPIC detectors are in, event pile up may be a problem. Pile up occurs when a source is so bright that incoming X-rays strike two neighboring pixels or the same pixel in the CCD more than once in a read-out cycle. In such cases the energies of the two events are in effect added together to form one event. Pile up and how to deal with it are discussed at length in §6.2.8 and §6.2.9, respectively, and users are strongly encouraged to refer to those sections. Briefly, we deal with it in PN Timing data essentially the same way as in Imaging data, that is, by using only single pixel events, and/or removing the regions with very high count rates, checking the amount of pile up, and repeating until it is no longer a problem.

In our example case, the source is far too faint to be piled up, but it is always a good idea to check anyway. To check for pile up,

- 1) Invoke *epatplot* in the SAS GUI.
- 2) In the “0” tab, in the “set” text area, enter the name of the event file that was made when the spectrum was extracted, `pn_filt_source.fits`. If you want to change the output file name to something other than the default, click the “useplotfile” box and enter the name in the “plotfile” text area. For this example, we will use `pn_epat.ps`.

- 3) In the “1” tab, set “withbackgroundset” to **yes**. In the “backgroundset” text area, enter the name of the background event file output when the background spectrum was extracted, `pn_filt_bkg.fits`.
- 4) Click “Run”.

The output is shown in Fig. 6.17.

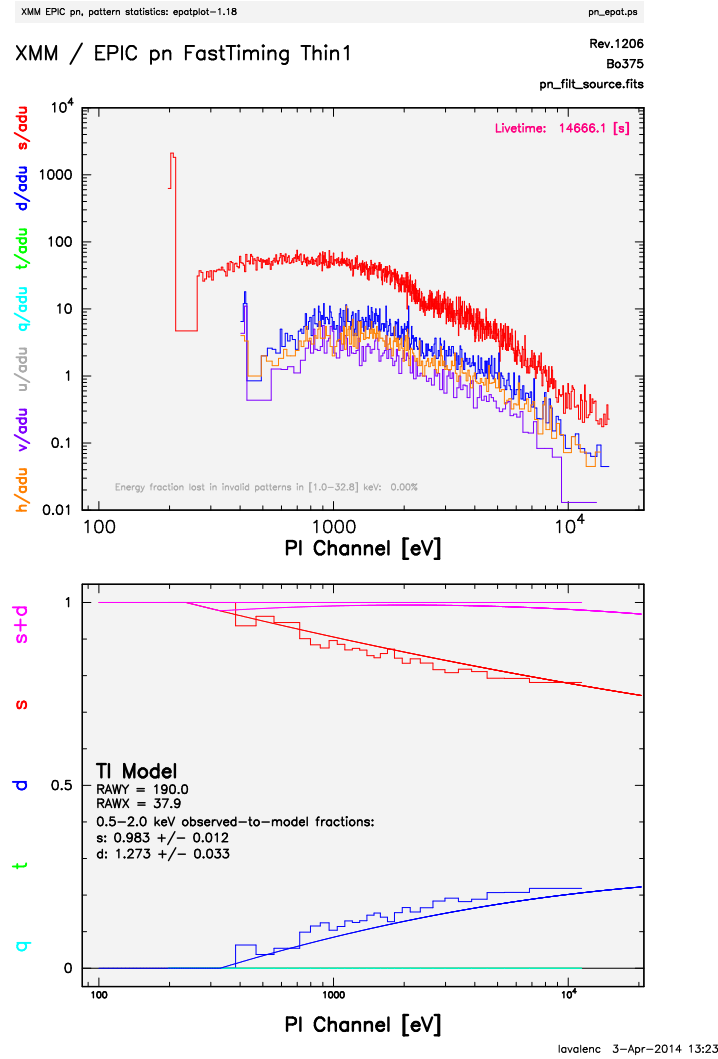


Figure 6.17: The output of *epatplot*.

6.6.8 My Timing Mode Data is Piled Up! Now What?

There are a couple ways to deal with pile up. First, you can use event file filtering procedures to include only single pixel events (`PATTERN==0`), as these events are less sensitive to pile up than other patterns.

You can also excise areas of high count rates, i.e., the boresight column and several columns to either side of it. (This is analogous to removing the inner-most regions of a source in Imaging data.) The spectrum can then be re-extracted and you can continue your analysis on the excised event file. As with Imaging data, it is recommended that you take an iterative approach: remove an inner region, extract a spectrum, check with *epatplot*, and repeat, each time removing a slightly larger region, until the model and observed pattern distributions agree.

For example, a bright source is shown in Fig. 6.18. We can extract the boresight column and several columns to either side:

- 1) Enter the filtering criteria in the “Selection Expression” area at the top of the *xmmselect* window: (FLAG==0) && (RAWX in [28:48])
- 2) Click the round button next the PI column on the *xmmselect* GUI.
- 3) Click on “OGIP Spectrum”.
- 4) In the “General” tab, check `keepfilteroutput` and `withfilteredset`. In the `filteredset` box, enter the name of the event file output, in this case, `pn_filt_source_WithBore.fits`.
- 5) In the “Spectrum” tab, set the file name and binning parameters for the spectrum. Confirm that `withspectrumset` is checked. Set `spectrumset` to the output name, `source_pi_WithBore.fits`. Confirm that `withspecranges` is checked. Set `specchannelmin` to 0 and `specchannelmax` to 20479.
- 6) Click “Run”.

We can do the same for columns with lower count rates to either side of the boresight by following the same steps as above but changing the filtering expression to (FLAG == 0) && (RAWX in [28:48]) &&! (RAWX in [34:41]). We will set the output spectrum file to be `source_pi_NoBore.fits`, and the output filtered event file to `pn_filt_source_NoBore.fits`, and check with *epatplot* as we go along until no pile up remains.

Be aware that reducing pile up in this way means that we will need to use a non-standard way to make the ancillary files (ARFs) for your spectrum! This is discussed further in §6.6.9. We will need the spectra of the full extraction area and the excised area, so we might as well get them now. We already have it for the full extraction area. For the excised area, we will follow the same steps as above, but change the filtering expression to (FLAG == 0) && (RAWX in [34:41]), the output spectrum will be `source_pi_Excised.fits` and the output event file will be `pn_filt_source_Excised.fits`.

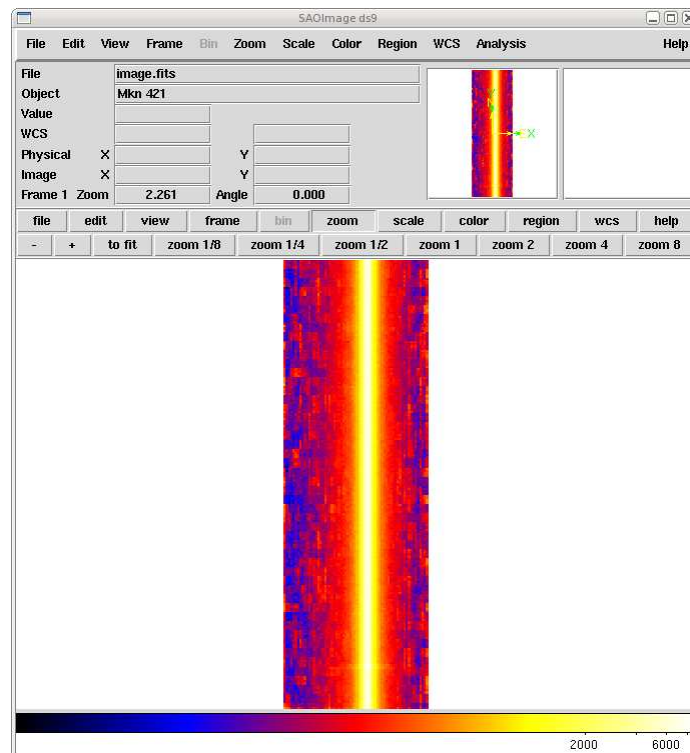


Figure 6.18: An image of a bright source in PN Timing mode.

6.6.9 Create the Photon Redistribution Matrix (RMF) and Ancillary File (ARF)

Making the RMF for PN data in Timing mode is exactly the same as in Imaging mode, which is demonstrated in §6.5.11. If the boresight was not excised, making the ARF is the very similar, as well.

To make the RMF,

- 1) Invoke the task *rmfgen* in the SAS GUI.
- 2) In the “Main” tab, set the **spectrumset** keyword to the spectrum file name, **source_pi.fits**. Set the **rmfset** keyword to the RMF file name, **source_rmf.fits**.
- 3) Click “Run”.

and to make the ARF,

- 1) Invoke the task *arfgen* in the SAS GUI.
- 2) In the “Main” tab, set the **arfset** parameter to the ARF file name, for example, **source_arf.fits**. Set the **spectrumset** parameter to the spectrum file name, **source_pi.fits**.
- 3) In the “Effects” tab, confirm that the **withbadpixcorr** box is checked. Set the **badpixlocation** keyword to the event file name from which the spectrum was extracted, **pn_filt.fits**.
- 4) In the “Calibration” tab, check the **withrmfset** box and set the **rmfset** keyword to the RMF file name, **source_rmf.fits**.
- 5) In the “Detector Map” tab, set the type of detector map type to “psf”.
- 6) Click “Run”.

However, *if you had to remove piled up sections, you will need to make an ARF for the full extraction area, another one for the piled up area, and then subtract the two to find the ARF for the non-piled regions.* We already have spectra for the full extraction area and the excised area, so we will use them to make the ARFs:

- 1) Invoke the task *arfgen* in the SAS GUI.
- 2) In the “Main” tab, set the **arfset** parameter to the output file name, **source_arf_WithBore.fits**. Set the **spectrumset** parameter to the spectrum file name, **source_pi_WithBore.fits**.
- 3) In the “Detector Map” tab, set the type of detector map type to “psf”.
- 4) Click “Run”.

We will repeat these four steps to make an ARF for **source_pi_Excised.fits**, which we will call **source_arf_Excised.fits**. Now we can subtract them. This is easily done on the command line:

```
addarf "source_arf_WithBore.fits source_arf_Excised.fits" "1.0 -1.0" source_arf.fits
```

At this point, the spectrum is ready to be analyzed, so we can prepare the spectrum for fitting (§6.7).

6.7 Prepare and Fit the Spectrum

With the source and background spectra now extracted and the RMF and ARF created, we will do some simple spectral fitting, using the example data from the EPIC Imaging mode (Lockman Hole, Obs ID 0123700101). SAS does not include fitting software, so HEASoft packages will be used, and all fitting tasks will be called from the command line.

Nearly all spectra will need to be binned for statistical purposes. The procedure *grppha* from FTOOL provides an excellent mechanism to do just that. The following commands not only group the source spectrum for Xspec but also associate the appropriate background and response files for the source.

- 1) On the command line, type:

```
grppha
```

and edit the parameters and file names as appropriate:

```
Please enter PHA filename[] mos1_pi.fits      ! input spectrum file name
Please enter output filename[] mos1_grp.fits   ! output grouped spectrum
GRPPHA[] chkey BACKFILE back_pi.fits         ! include the background spectrum
GRPPHA[] chkey RESPFILE mos1_rmf.fits         ! include the RMF
GRPPHA[] chkey ANCRFILE mos1_arf.fits         ! include the ARF
GRPPHA[] group min 25                        ! group the data by 25 counts/bin
GRPPHA[] exit
```

Next, use Xspec to fit the spectrum.

- 1) On the command line, type:

```
xspec
```

and edit the parameters and file names as appropriate:

```
XSPEC> data mos1_grp.fits      ! input data
XSPEC> ignore 0.0-0.2,6.6-**   ! ignore unusable energy ranges, in keV
                                ! set a range appropriate for the data
XSPEC> model wabs(pow+pow)     ! set spectral model to two absorbed power laws
1:wabs:nH> 0.01                ! set model absorption column density to 1.e20
2:powerlaw:PhoIndex> 2.0       ! set the first model power law index to -2.0
3:powerlaw:norm>               ! default model normalization
4:powerlaw:PhoIndex> 1.0       ! set the second model power law index to -1.0
5:powerlaw:norm>               ! default model normalization
wabs:nH> 0.01                  ! set model absorption column density to 1.e20
renorm                         ! renormalize the model spectrum
XSPEC> fit                     ! fit the model to the data
XSPEC> setplot device /xw      ! set the plot device
XSPEC> setplot energy          ! plot energy along the X axis
XSPEC> plot ldata ratio        ! plot two panels with the log of the data and
                                ! the data/model ratio values along the Y axes
XSPEC> exit                    ! exit Xspec
Do you really want to exit? (y) y
```

Figure 6.19 shows the fit to the spectrum.

6.8 Timing Analysis

The subpackage Xronos in ftools can be used in general timing analysis of EPIC data (any mode). This section will demonstrate some basic timing analysis using it and the example PN Timing data, Bol 375 (Observation ID 0403530301). More information about Xronos can be found through the ftools website: <https://heasarc.gsfc.nasa.gov/lheasoft/ftools/xronos.html>.

We will assume that an appropriate light curve, named `pn_ltcrv.fits`, has been created, with the parameters `timebinsize` set to 1 and `makeratecolumn` set to `no`. Details on how to make such a light curve can be found in §6.3.2 (for the command line) or §6.6.2 (for the GUI). Note that for many Xronos tasks, though QDP may plot the results, the output file is saved as a FITS file.

First, a light curve can be binned using the Xronos command `lcurve`. It will be plotted automatically using QDP; just type “quit” or “exit” to end the session:

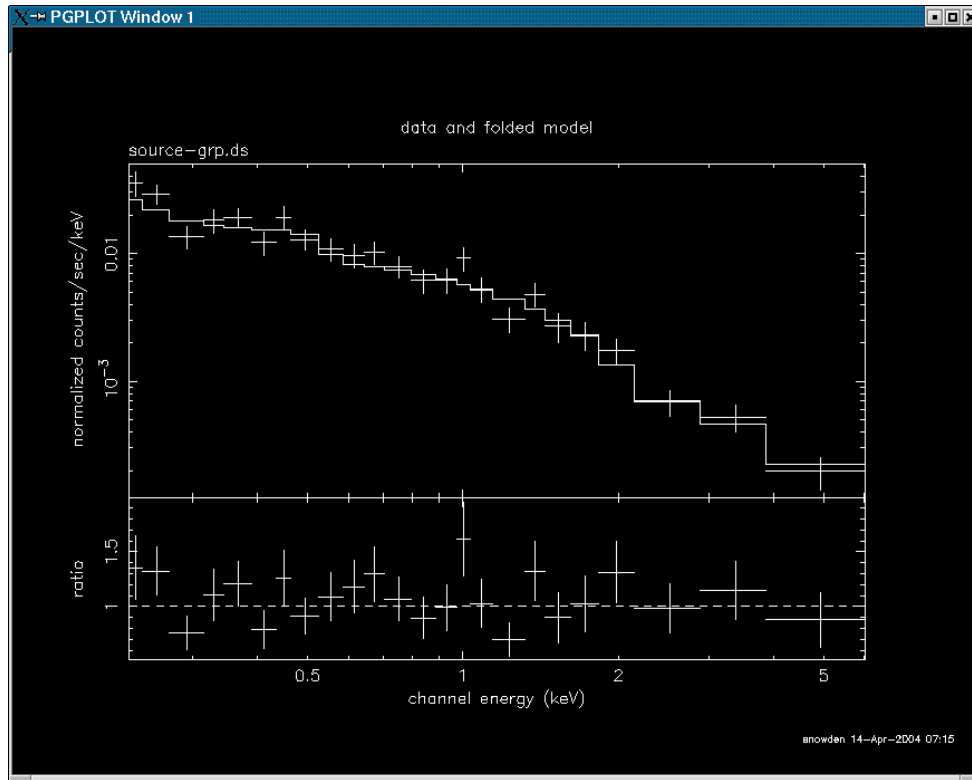


Figure 6.19: The fitted spectrum of the Lockman Hole source.

```
lcurve nser=1 cfile1=pn_ltcrv.fits window=- dtnb=INDEF
      nbint=INDEF outfile=lightcurve.fits plot=yes plotdev=/xw
```

where

nser – number of time series
cfile1 – filename first series
window – name of window file (if a subset of the time series is required)
dtb – bin size (time; set to INDEF for default)
nbint – number of bins per interval (set to INDEF for default)
outfile – output file name (FITS format light curve)
plot – plot flag
plotdev – device for plotting output

The output is shown in Fig. 6.20.

The power spectrum density can be found using the command *powspec*. As with *lcurve*, the plot is shown automatically in QDP, and “quit” or “exit” will end the session.

```
powspec cfile1=pn_ltcrv.fits window=- dtnb=10 nbint=200
      nintfm=INDEF rebin=5 plot=yes plotdev=/xw outfile=power.fits
```

where

cfile1 – filename first series
window – name of window file (if a subset of the time series is required)
dtb – bin size (time; set to INDEF for default)

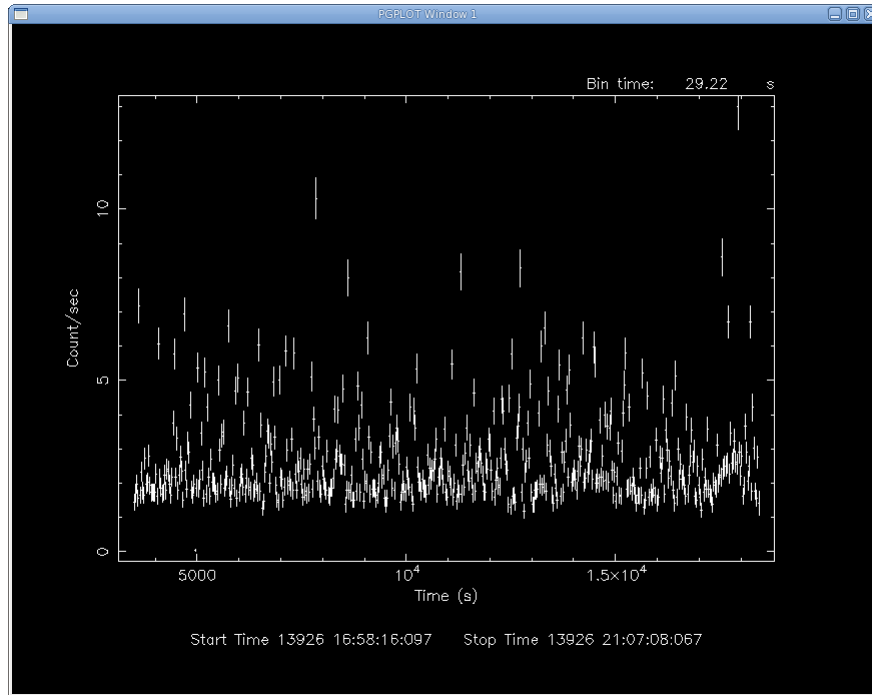


Figure 6.20: The light curve as seen in Xronos.

nbint – number of bins per interval (set to INDEF for default)
nintfm – number of intervals in each power spectrum (set to INDEF for default)
rebin – rebin factor for power spectrum (0 for no rebinning)
plot – plot flag
plotdev – device for plotting
outfile – output file name

The output is shown in Fig. 6.21.

We can look for periodicities in the time series using the command *efsearch*. Once again, this will automatically produce a plot with QDP.

```
efsearch cfile1=pn1lcrv.fits window=- seepoch=INDEF dper=20 nphase=10
        nbint=INDEF nper=100 dres=INDEF plot=yes plotdev=/xw outfile=efsearch.fits
```

where

cfile1 – filename first series
window – name of window file (if a subset of the time series is required)
seepoch – value for epoch used for phase zero when folding the time series
dper – value for the period used in the folding
nphase – number of phases per period
nbint – number of bins per interval
nper – number of sampled periods during search
dres – sampling resolution of search
plot – plot flag
plotdev – device for plotting
outfile – output file name

We can calculate the auto correlation for an input time series using the command *autocor*.

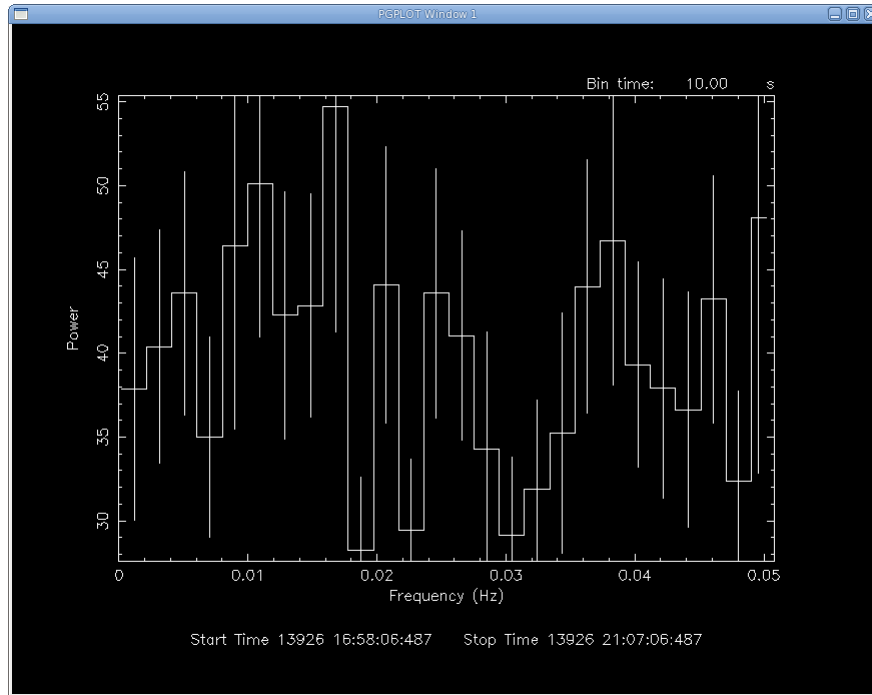


Figure 6.21: The light curve as seen in Xronos.

```
autocor cfile1=pn_ltrcv_xronos.fits window=- dtnb=10 nbint=100 nintfm=INDEF
        rebin=0 plot=yes plotdev=/xw outfile=auto.fits
```

where

```
cfile1 - filename first series
window - name of window file (if a subset of the time series is required)
dtnb - bin size (time)
nbint - number of bins per interval
nintfm - number of intervals to be summed in each autocorrelation function
rebin - rebin factor for autocorrelation function (0 for no rebinning)
plot - plot flag
plotdev - device for plotting
outfile - output file name
```

Numerous statistics can be calculated for an input time series using the command *lcstats*. The output is written to the screen by default, but by appending “> output.txt”, the output is sent to the text file *output.txt* instead.

```
lcstats cfile1=pn_ltrcv_xronos.fits window=- dtnb=INDEF nbint=INDEF > output.txt
```

where

```
cfile1 - filename first series
window - name of window file
dtnb - integration time (binning)
nbint - number of bins
```

6.9 Once More, This Time with Feeling and Ftools

Most of the data extraction described in the previous sections can be done equally well in Ftools, and will be illustrated here using *fselect* and *Xselect*. Note that the HEASoft package is incorporated into SAS and so if SAS is operational, *fselect* and *Xselect* will be available. Keith Arnaud is responsible for the *XMM-Newton*-specific tools mentioned below, which he describes at:

<http://lheawww.gsfc.nasa.gov/users/kaa/xselect/xmm.html>

- 1) Filter the event file using the *xmmclean perl* script provided by Arnaud at the HTML page above by typing the following:

```
fselect mos1.fits mos1-filt.fits "FLAG ==0 && TIME <= 73227600)
&&!(TIME in [73221920:73223800]) && PATTERN <= 12
&& PI <= 12000 && PI >= 200"
```

- 2) Invoke an *Xselect* session.

```
xselect
```

Enter a session name or default with a carriage return.

- 3) Read in the event list.

```
read events mos1-filt.fits
```

Enter the directory containing the event file; enter **yes** to reset the mission.

- 4) Create and plot an image (this will spawn a *ds9* window).

```
extract image
plot image
```

- 5) Create and plot a light curve (this will spawn a *Pgplot* window).

```
extract curve
plot curve
```

- 6) Filter the data. This can be done in one of two ways: using the cursor and light curve plot, or by using a threshold intensity.

- a) To filter on time using the cursor and light curve plot, type

```
filter time cursor
```

and follow the instructions.

- Enter “quit” at the PLT prompt.
- Right-click at the start and end points of the time intervals to keep.
- When done entering intervals, enter “x” on the keyboard

- b) To filter on time using a threshold intensity, type

```
filter time cursor range
```

where *range* is the filter range, e.g., 0.01–3.0.

- 7) Create the extraction region for the source.

- a) Display the filtered image output from step 6 with *ds9*.

- b) Create a region on the *ds9* window:

- In the *ds9* window pull down the **Region** menu and set 1) the **File Format** to DS9/Funtools, 2) the **File Coordinate System** to Equatorial J2000, and 3) the **Region Coordinate System** to Degrees

- c) Adjust the region to be appropriate for the source of interest

- d) Under the region menu select the **Save Regions** option
 - e) Save the region as a file (e.g., **ds9-source.reg**)
- 8) Create an annulus extraction region for the background.
- a) If necessary, resize the existing region to be appropriate for the inner annulus radius
 - b) Pull down the **Region** menu and select **Exclude** under **Properties**
 - c) Create a second region on the *ds9* window
 - d) Adjust the region to be appropriate for the outer boundary of the annulus
 - e) Pull down the **Region** menu and select **Include** under **Properties**
 - f) Make sure that the outer annulus is “in front” by selecting the **Move to Front** option under the **Region** menu.
 - g) Under the region menu select the **Save Regions** option
 - h) Save the region (e.g., **ds9-back.reg**)
- 9) Filter the data using the source region.
- **filter region ds9-source.reg**
- 10) Extract, plot, save the spectrum from the source region and create RMF and ARF files.
- ```
extract spectrum
plot spectrum
save spectrum resp=yes
```
- The **resp=yes** runs the *perl* script **xsl\_xmm\_epic\_makeresp** which is available from Arnaud’s web page above
- Enter a file name for the spectrum, e.g., **mos1-source.pi**
  - Bin the data (i.e., enter **yes** at the query)
- 11) Filter the data using the background region.
- a) First remove the source filter expression: **clear region all**
  - b) Now invoke the background filter: **filter region ds9-back.reg**
- 12) Extract, plot, and save the spectrum from the background region.
- ```
– extract spectrum
– plot spectrum
– save spectrum
```
- Enter a file name for the spectrum, e.g., **mos1-back.pi**
 - Bin the data (i.e., enter **yes** at the query)
- 13) Extract, plot, and save the light curve from the region.
- a) First remove the source filter expression: **clear region all**
 - b) Now invoke the source filter: **filter region ds9-source.reg**
 - c) And extract the light curve.
- ```
extract curve binsize=1000 phalcut_t=300 phahcut_t=10000
```
- use **binsize=1** to create a light curve for timing analysis
  - use **phalcut\_t** to set the lower energy bound for the light curve
  - use **phahcut\_t** to set the upper energy bound for the light curve
- ```
plot curve
save curve
```
- Enter a file name for the light curve

From this point follow the procedures in § 6.7 for spectral analysis and § 6.8 for temporal analysis.

Chapter 7

An RGS Data Processing and Analysis Primer

Before beginning this chapter please consult the “watchout” page at the SOC:

<http://xmm.esac.esa.int/sas/current/watchout/>

This web site discusses current and past SAS bugs and analysis issues.

Many files are associated with an RGS dataset, and it is easy to be overwhelmed. The `INDEX.HTM` file, and links therein, are viewable with a web browser and will help you navigate the dataset. The different types of files are discussed in more detail in Chapter 1.

As ever, it is strongly recommended that you keep all reprocessed data in its own directory! SAS places output files in whichever directory it is in when a task is called. Throughout this primer, it is assumed that the Pipeline Processed data are in the PPS directory, the ODF data (with upper case file names, and uncompressed) are in the directory ODF, the reprocessing and analysis is taking place in the PROC directory, and the CCF data are in the directory CCF.

If you have just received your data from the SOC, it has been processed with the most recent version of SAS, and you should not need to repipeline it (though no harm is done if you do); you need only to gunzip the files and prepare the data for processing (see §5). However, it is very likely that you will want to filter your data; in this case, you will need to reprocess it in order to determine the appropriate filters. Therefore, we recommend that you rerun the pipeline regardless of the age of your dataset.

But if you decide that reprocessing is unnecessary, you need only to gunzip the files and rename event files for easier handling. For example, for the RGS1 event list,

```
cp PPS/PiiiiijjkkR11EVENLinmmm.FTZ PROC/r1_evt.fits
```

where

`iiiiijjkk` – observation number
`1` – scheduled (S) or unscheduled (U) obseravtion
`n` – spectral order number
`mmm` – source number

As noted in Tables 3.2 and 3.3 you can view images of your data. While the zipped FITS files may need to be unzipped before display in *ds9* (depending on the version of *ds9*), they can be displayed when zipped using *fv*. As usual, there are some HTML products to help you inspect the data. These have file names of the form:

```
PPiiiiijjkkAAAAA000_0.HTM
```

where

`iiiiijjkk` – Observation number
`jj` – observation ID - target number in proposal
`kk` – observation ID - observation number for target
`AAAAAA` – Group ID (see Table 3.2)

You will find a variety of RGS-specific files in *XMM-Newton* data sets. Generally there are two of each because there are two RGS instruments. Table 3.3 lists typical file names, their purpose, the file format, and a list of tools that will enable the user to inspect their data. Remember that the `INDEX.HTM` file will help you navigate.

Various analysis procedures are demonstrated using the Mkn 421 dataset, ObsID 0153950701, which definitely needs to be repipelined. The following procedures are applicable to all XMM-Newton datasets, so it is not required that you use this particular dataset; any observation should be sufficient.

The next two sections (§7.1 and §7.2) will deal with pipelining and filtering RGS data on the command line, respectively. Users who prefer to use the GUI should skip ahead to find the demonstrations for the same tasks with the GUI (§7.3 and §7.4).

7.1 Rerun the Pipeline on the Command Line

We assume that the data was prepared and environment variables were set according to §5. In the window where SAS was initialized, in your “processing directory” PROC, run the task *rgsproc*:

```
rgsproc orders='1 2' bkgcorrect=no withmlambdacolumn=yes spectrumbinning=lambda
```

where

```
orders – dispersion orders to extract
bkgcorrect – subtract background from source spectra?
withmlambdacolumn – include a wavelength column in the event file product
spectrumbinning – accumulate the spectrum either in wavelength or beta space
```

Note the last keyword, `spectrumbinning`. If you want to merge data from the same orders in RGS1 and RGS2, keep it at the default value `lambda`. Merging spectra is discussed in §7.2.6.

This takes several minutes, and outputs 12 files per RGS, plus 3 general use FITS files. At this point, renaming files to something easy to type is a good idea.

```
ln -s *R1*EVENLI*FIT r1_evt1.fits
ln -s *R2*EVENLI*FIT r2_evt1.fits
```

7.1.1 Potentially useful tips for using the pipeline

The pipeline task, *rgsproc*, is very flexible and can address potential pitfalls for RGS users. In §7.1, we used a simple set of parameters with the task; if this is sufficient for your data (and it should be for most), feel free to skip to §7.2, where data filters are discussed. In the following subsections, we will look at the cases of a nearby bright optical source, a nearby bright X-ray source, and a user-defined source.

7.1.2 A Nearby Bright Optical Source

With certain pointing angles, zeroth-order optical light may be reflected off the telescope optics and cast onto the RGS CCD detectors. If this falls on an extraction region, the current energy calibration will require a wavelength-dependent zero-offset. Stray light can be detected on RGS DIAGNOSTIC images taken before, during and after the observation. This test, and the offset correction, are not performed on the data before delivery. To check for stray light and apply the appropriate offsets, type

```
rgsproc orders='1 2' bkgcorrect=no calcoffsets=yes withoffsethistogram=no
```

where the parameters are as described in §7.1 and

```
calcoffsets – calculate PHA offsets from diagnostic images
withoffsethistogram – produce a histogram of uncalibrated excess for the user
```

7.1.3 A Nearby Bright X-ray Source

In the example above, it is assumed that the field around the source contains sky only. Provided a bright background source is well-separated from the target in the cross-dispersion direction, a mask can be created that excludes it from the background region. Here the source has been identified in the EPIC images and its coordinates have been taken from the EPIC source list which is included among the pipeline products. The bright neighboring object is found to be the third source listed in the sources file. The first source is the target:

```
rgsproc orders='1 2' bkgcorrect=no withepicset=yes
epicset=Piiiiijjkaabl11EMSRLInmmm.FTZ exclsrcsexpr='INDEX==1&&INDEX==3'
```

where the parameters are as described in §7.1 and

withepicset – calculate extraction regions for the sources contained in an EPIC source list
epicset – name of the EPIC source list, such as generated by *emldetect* or *eboxdetect* procedures
exclsrcsexpr – expression to identify which source(s) should be excluded from the background extraction region

7.1.4 User-defined Source Coordinates

If the true coordinates of an object are not included in the EPIC source list or the science proposal, the user can define the coordinates of a new source by typing:

```
rgsproc orders='1 2' bkgcorrect=no withsrc=yes srclabel=Mkn421 srcstyle=radec
srcra=166.113808 srcdec=+38.208833
```

where the parameters are as described in §7.1 and

withsrc – make the source be user-defined
srclabel – source name
srcstyle – coordinate system in which the source position is defined
srcra – the source's right ascension in decimal degrees
srcdec – the source's declination in decimal degrees

7.2 Examine and Analyze the Data on the Command Line

Since the event files are current, we can proceed with some simple analysis demonstrations, which will allow us to generate filters. Rememer that all tasks should be called from the window where SAS was initiated, and that tasks place output files in whatever directory you are in when they are called.

7.2.1 Create and Display an Image

Two commonly-made plots are those showing PI vs. BETA_CORR (also known as “banana plots”) and XDSP_CORR vs. BETA_CORR.

To create images on the command line, type

```
evselect table=r1_evt1.fits:EVENTS withimageset=yes
imageset=pi.bc.fits xcolumn=BETA_CORR ycolumn=PI
imagebinning=imageSize ximagesize=600 yimagesize=600
```

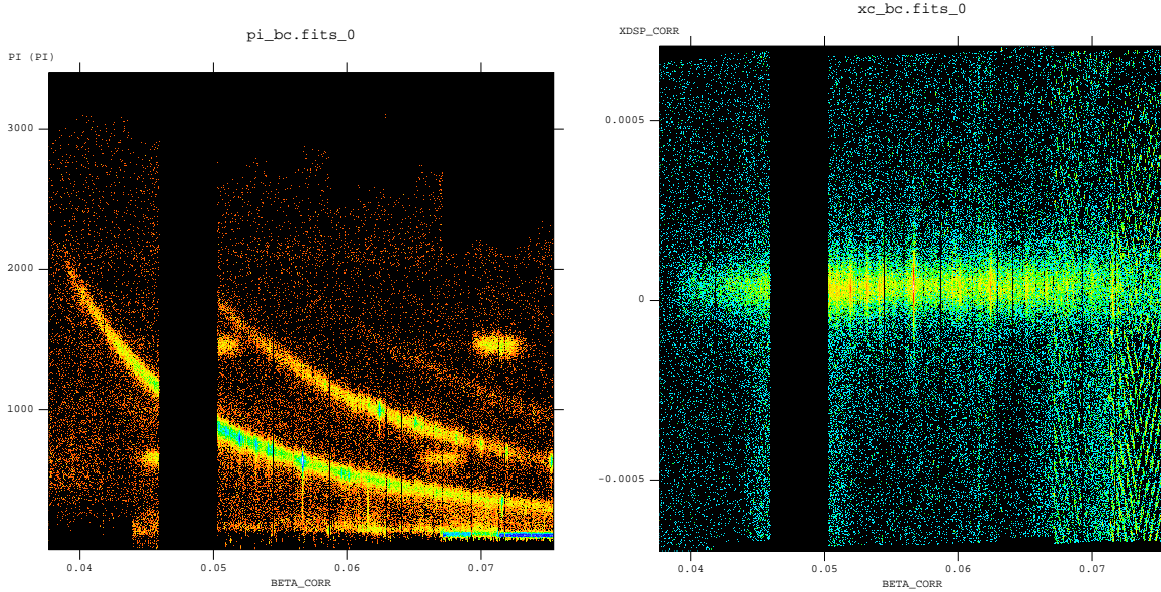
where

table – input event table
withimageset – make an image
imageset – name of output image
xcolumn – event column for X axis
ycolumn – event column for Y axis
imagebinning – form of binning, force entire image into a given size or bin by a specified number of pixels

ximagesize – output image pixels in X
yimagesize – output image pixels in Y

Plots comparing BETA_CORR to XDSP_CORR may be made in a similar way. The output files can be viewed by using a standard FITS display, such as *ds9* (see Figure 7.1).

Figure 7.1: Plots of PI vs. BETA_CORR (left) and XDSP vs. BETA_CORR (right).



7.2.2 Create and Display a Light Curve

The background is assessed through examination of the light curve. We will extract a region, CCD9, that is most susceptible to proton events and generally records the least source events due to its location close to the optical axis. Also, to avoid confusing solar flares for source variability, a region filter that removes the source from the final event list should be used. The region filters are kept in the source file product P*SRCLI*.FIT.

More experienced users should be aware that with SAS 13, the *SRCLI* file's **column information changed**. *rgsproc* now outputs an M_LAMBDA column instead of BETA_CORR, and M_LAMBDA should be used to generate the light curve. (The *SRCLI* file that came with the PPS products still contains a BETA_CORR column if you prefer to use that instead.)

To create a light curve, type

```
evselect table=r1_evt1.fits withrateset=yes rateset=r1_ltcrv.fits
maketimecolumn=yes timebinsize=100 makeratecolumn=yes
expression=
' (CCDNR==9)&&(REGION(P0153950701R1S001SRCLI_0000.FIT:RGS1_BACKGROUND,M_LAMBDA,XDSP_CORR)) '
```

where

table – input event table
withrateset – make a light curve
rateset – name of output light curve file
maketimecolumn – control to create a time column
timebinsize – time binning (seconds)
makeratecolumn – control to create a count rate column, otherwise a count column will be created
expression – filtering criteria

The output file `r1_ltcrv.fits` can be viewed using *dsplot*:

```
dsplot table=r1_ltcrv.fits x=TIME y=RATE &
```

where

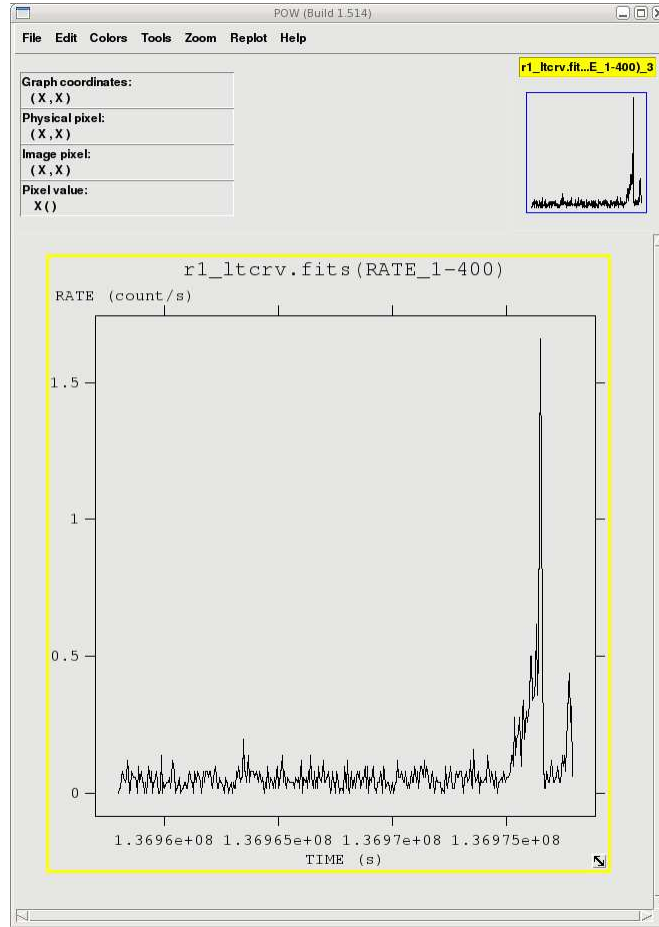
`table` – input event table

`x` – column for plotting on the X axis

`y` – column for plotting on the Y axis

The light curve is shown in Figure 7.2.

Figure 7.2: Background event rate from the RGS1 CCD9 chip. The flares are solar events. The time units are elapsed mission time.



7.2.3 Generating the Good Time Interval (GTI) File

Examination of the lightcurve shows that there is a noisy section at the end of the observation, after 1.36975e8 seconds, where the count rate is well above the normal background count rate of ~ 0.05 count/second. There are two procedures that make the GTI file (*gtibuild* and *tabgtigen*) that, when applied to the event file in another run of *rgsproc*, will excise these sections.

The first method, using *gtibuild*, requires a text file as input. In the first two columns, refer to the start and end times (in seconds) that you are interested in, and in the third column, indicate with either a + or - sign whether that region should be kept or removed. Each good (or bad) time interval should get its own line. In the example case, we would write in our ASCII file (named `gti.txt`):

```
1.36958e8 1.36975e8 +
```

and proceed to the SAS task *gtibuild*:

```
gtibuild file=gti.txt table=gti.fits
```

where

file – input text file
table – output gti table

Alternatively, we can make the GTI file with *tabgtigen* and filter for RATE (though we could just as easily filter on TIME):

```
tabgtigen table=r1_ltcv.fits gtiset=gti.fits expression='(RATE<0.2)'
```

where

table – the lightcurve file
gtiset – output gti table
expression – the filtering criteria. Since the nominal count rate is 0.05 about count/sec, we have set the upper limit to 0.2 count/sec.

7.2.4 Applying the GTI

Now that we have GTI file, we can apply it to the event file by running *rgsproc* again. *rgsproc* is a complex task, running several steps, with five different entry and exit points. It is not necessary to rerun all the steps in the procedure, only the ones involving filtering.

To apply the GTI to the event file, type

```
rgsproc orders='1 2' auxgtitables=gti.fits bkgcorrect=no  
withmlambdacolumn=yes entrystage=3:filter finalstage=5:fluxing
```

where

orders – spectral orders to be processed
auxgtitables – gti file in FITS format
bkgcorrect – subtract background from source spectra?
withmlambdacolumn – include a wavelength column in the event file product
entrystage – stage at which to begin processing
finalstage – stage at which to end processing

7.2.5 Creating the Response Matrices (RMFs)

Response matrices (RMFs) are not provided as part of the pipeline product package, so you must create your own before analyzing data. The task *rgsproc* generates a response matrix automatically, but as noted in §7.1.4, the source coordinates are under the observer's control. The source coordinates have a profound influence on the accuracy of the wavelength scale as recorded in the RMF that is produced automatically by *rgsproc*, and each RGS instrument and each order will have its own RMF.

Making the RMF is easily done with the package *rgsrmfgen*. Please note that, unlike with EPIC data, it is not necessary to make ancillary response files (ARFS).

To make the RMFs, type

```
rgsrmfgen spectrumset=P0153950701R1S001SRSPEC1001.FIT rmfset=r1_o1_rmf.fits  
evlist=r1_evt1.fits emin=0.4 emax=2.5 rows=5000
```

where

spectrumset – spectrum file
evlist – event file
emin – lower energy limit of the response file
emax – upper energy limit of the response file
rows – number of energy bins; this should be greater than 3000
rmfset – output FITS file

At this point, the spectra can be analyzed. If you wish, skip the discussion on combining spectra (§7.2.6) and go straight to fitting the spectrum (§7.5.)

7.2.6 Combining Spectra

Spectra from the same order in RGS1 and RGS2 can be safely combined to create a spectrum with higher signal-to-noise if they were reprocessed using *rgsproc* with `spectrumbinning=lambd`, as we did in §7.1 (this also happens to be the default). The task *rgscombine* also merges response files and background spectra. When merging response files, be sure that they have the same number of bins. For this example, we assume that RMFs were made for order 1 in both RGS1 and RGS2.

To merge RGS1 and RGS2 spectra, type

```
rgscombine pha='P0153950701R1S001SRSPEC1001.FIT P0153950701R2S002SRSPEC1001.FIT'
           rmf='r1_o1_rmf.fits r2_o1_rmf.fits'
           bkg='P0153950701R1S001BGSPEC1001.FIT P0153950701R2S002BGSPEC1001.FIT'
           filepha='r12_o1_srspec.fits' filermf='r12_o1_rmf.fits'
           filebkg='r12_o1_bgspec.fits' rmfgrid=5000
```

where

```
pha – list of spectrum files
rmf – list of response matrices
bkg – list of background spectrum files
filepha – output merged spectrum
filermf – output merged response matrix
filebkg – output merged background spectrum
rmfgrid – number of energy bins; should be the same as the input RMFs
```

The spectra are ready for analysis, so skip ahead to prepare the spectrum for fitting (§7.5).

7.3 Rerun the Pipeline with the GUI

We assume that the data was prepared and environment variables were set according to §5. In the window where SAS was initialized, in your “processing directory” PROC, invoke the SAS GUI (see Figure 6.10).

On the command line, type

```
sas &
```

A window will appear with two panels. The upper one will contain task names, what group they belong to (such as utility, epic, timing, calibration, etc.) and a short description of each. The lower one will contain information about environment variables, and as tasks are invoked, feedback from the tasks.

From the upper window, select *rgsproc*. As with all SAS tasks in the GUI, double-clicking the task will bring up pop-up windows that will allow you to change the parameters. For *rgsproc*,

- 1) In the “global” tab, verify that “orders” is set to 1 2 and “spectrumbinning” is set to `lambd`. In the “angles” tab, verify that “withmlambdacolumn” is set to `yes`. In the “spectra” tab, under the “rgsspectrum” sub-tab, verify that “bkgcorrect” is set to `no`.
- 2) Click “Run”.

This takes several minutes, and outputs 12 files per RGS, plus 3 general use FITS files. At this point, renaming files to something easy to type is a good idea. In the following sections, we will refer to the RGS1 event file, P0134520301R1S001EVENLI0000.FIT as `r1_evt1.fits`.

7.3.1 Potentially useful tips for using the pipeline

The pipeline task, *rgsproc*, is very flexible and can address potential pitfalls for RGS users. In §7.3, we used a simple set of parameters with the task; if this is sufficient for your data (and it should be for most), feel free to skip to §7.4, where data filters are discussed. In the following subsections, we will look at the cases of a nearby bright optical source, a nearby bright X-ray source, and a user-defined source.

7.3.2 A Nearby Bright Optical Source

With certain pointing angles, zeroth-order optical light may be reflected off the telescope optics and cast onto the RGS CCD detectors. If this falls on an extraction region, the current energy calibration will require a wavelength-dependent zero-offset. Stray light can be detected on RGS DIAGNOSTIC images taken before, during and after the observation. This test, and the offset correction, are not performed on the data before delivery.

To check for stray light and apply the appropriate offsets,

- 1) Call *rgsproc*.
- 2) In the “global” tab, verify that “orders” is set to 1 2 and “spectrumbinning” is set to **lambda**. In the “angles” tab, verify that “withmlambdacolumn” is set to **yes**. In the “spectra” tab, under the “rgsspectrum” sub-tab, verify that “bkgcorrect” is set to **no**.
- 3) In the “events” tab, under the “rgsoffsetcalc” sub-tab, set “calcoffsets” to **yes** and “withoffsethistogram” to **no**.
- 4) Click “Run”.

7.3.3 A Nearby Bright X-ray Source

In the example above, it is assumed that the field around the source contains sky only. Provided a bright background source is well-separated from the target in the cross-dispersion direction, a mask can be created that excludes it from the background region. Here the source has been identified in the EPIC images and its coordinates have been taken from the EPIC source list which is included among the pipeline products. The bright neighboring object is found to be the third source listed in the sources file. The first source is the target:

- 1) Call *rgsproc*.
- 2) In the “global” tab, verify that “orders” is set to 1 2 and “spectrumbinning” is set to **lambda**. In the “angles” tab, verify that “withmlambdacolumn” is set to **yes**. In the “spectra” tab, under the “rgsspectrum” sub-tab, verify that “bkgcorrect” is set to **no**.
- 3) In the “events” tab, under the “rgssources” sub-tab, set “withepicset” to **yes** and “epicset” to the name of the EPIC source list as generated by *emldetect* or *eboxdetect*. In the “spectra” tab, under the “rgsregions” sub-tab, set “exclsrcexpr” to **INDEX==1&&INDEX==3**.
- 4) Click “Run”.

7.3.4 User-defined Source Coordinates

If the true coordinates of an object are not included in the EPIC source list or the science proposal, the user can define the coordinates of a new source.

- 1) Call *rgsproc*.
- 2) In the “global” tab, verify that “orders” is set to 1 2 and “spectrumbinning” is set to **lambda**. In the “angles” tab, verify that “withmlambdacolumn” is set to **yes**. In the “spectra” tab, under the “rgsspectrum” sub-tab, verify that “bkgcorrect” is set to **no**.
- 3) In the “events” tab, under the “rgssources” sub-tab, set “withsrc” to **yes**. Be sure that the box beneath “withsrc” is toggled to **radec**. Next to “srcra”, enter the RA in decimal degrees, **166.113808** and next to “srcdec”, enter the declination (again in decimal degrees), **+38.208833**. Next to “srclabel”, enter **Mkn421**.
- 4) Click “Run”.

7.4 Examine and Analyze the Data with the GUI

Since the event files are current, we can proceed with some simple analysis demonstrations, which will allow us to generate filters. Remember that all tasks should be called from the window where SAS was initiated, and that tasks place output files in whatever directory you are in when they are called.

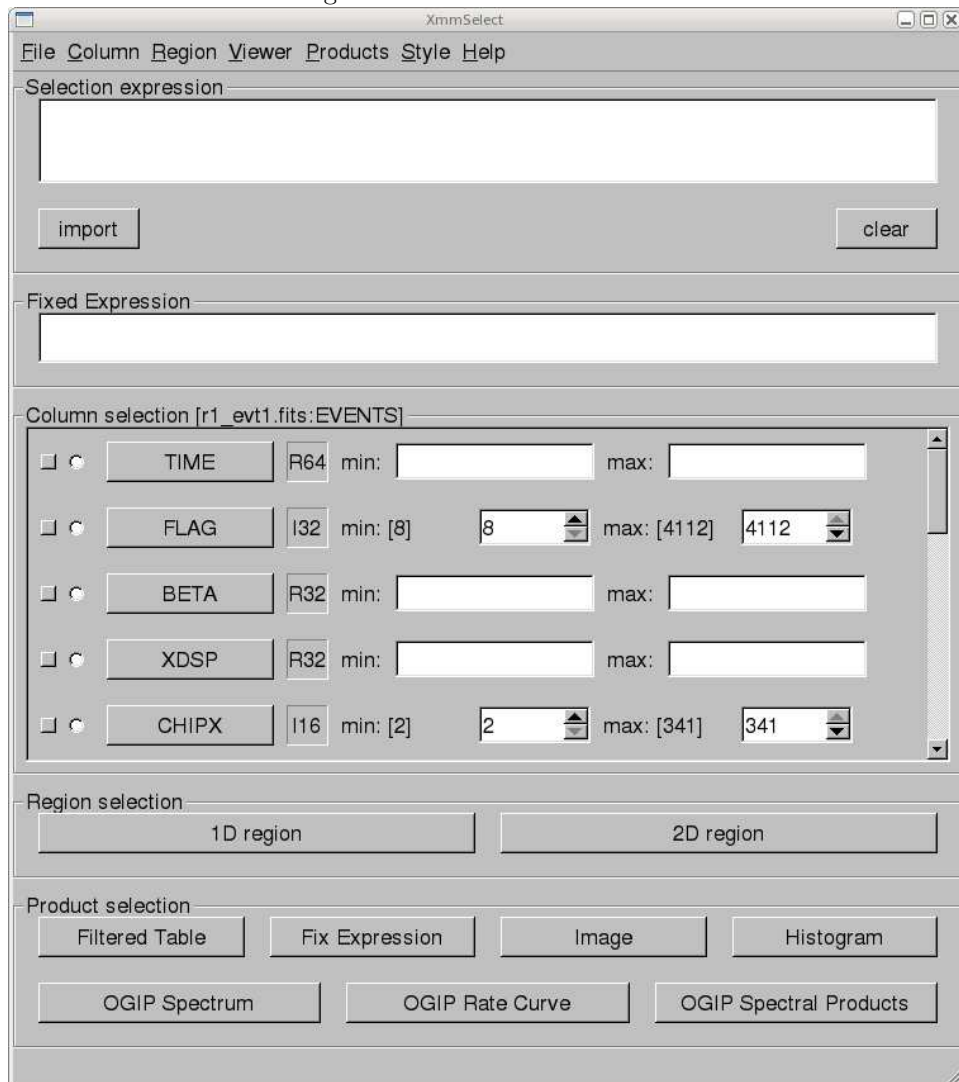
7.4.1 An Introduction to *xmmselect*

The task *xmmselect* is used for many procedures in the GUI. Like all tasks, it can easily be invoked by starting to type the name and pressing enter when it is highlighted.

When *xmmselect* is invoked a dialog box will first appear requesting a file name. You can either use the browser button or just type the file name in the entry area, “r1_evts1.fits:EVENTS” in this case. To use the browser, select the file folder icon; this will bring up a second window for the file selection. Choose “r1_evts1.fits”, then the “EVENTS” extension in the right-hand column, and click “OK”. The directory window will then disappear and you can click “Run” on the selection window.

When the file name has been submitted the *xmmselect* GUI (see Figure 7.3) will appear. (Later, when loading an event file that has been filtered, the filters that have been applied to that point will appear in the selection expression area.)

Figure 7.3: The *xmmselect* GUI.



7.4.2 Create and Display an Image

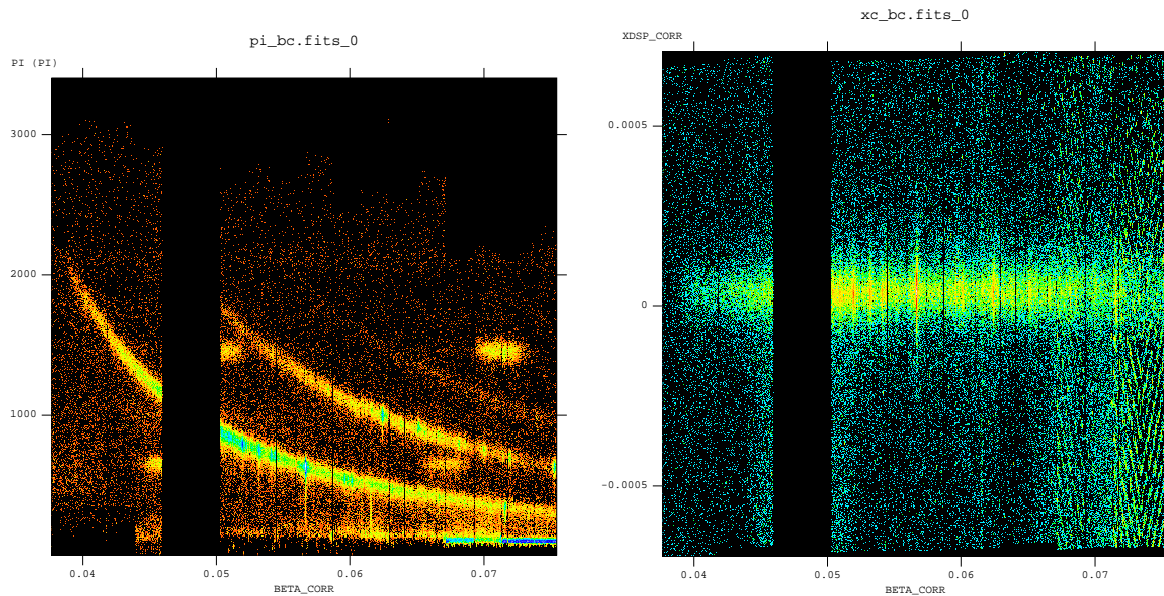
Two commonly-made plots are those showing PI vs. BETA_CORR (also known as “banana plots”) and XDSP_CORR vs. BETA_CORR.

To create images,

- 1) Check the square boxes to the left of the “BETA_CORR” and “PI” entries.
- 2) Click on the “Image” button near the bottom of the page. This brings up the *evselect* GUI.
- 3) Click on the “Image” tab in the *evselect* GUI.
- 4) Confirm that the `withimageset` box is checked.
- 5) In the imageset box, change the output image name from `image.ds` to something descriptive, in this case, `pi_bc.fits`.
- 6) Click on the “Run” button on the lower left corner of the *evselect* GUI.

Different binnings and other selections can be invoked by accessing the “Image” tab at the top of the GUI. The default settings are reasonable, however, for a basic image. The resultant image is automatically displayed using *ds9*. Similarly, plots can be made comparing BETA_CORR to XDSP_CORR. These two example plots can be seen in Figure 7.4.

Figure 7.4: Plots of PI vs. BETA_CORR (left) and XDSP vs. BETA_CORR (right).



7.4.3 Create and Display a Light Curve

The background is assessed through examination of the light curve. We will extract a region, CCD9, that is most susceptible to proton events and generally records the least source events due to its location close to the optical axis. Also, to avoid confusing solar flares for source variability, a region filter that removes the source from the final event list should be used. The region filters are kept in the source file product `P*SRCLI*.FIT`. (For our example data, this would be `P0134520301R1S001SRCLI_0000.FIT`).

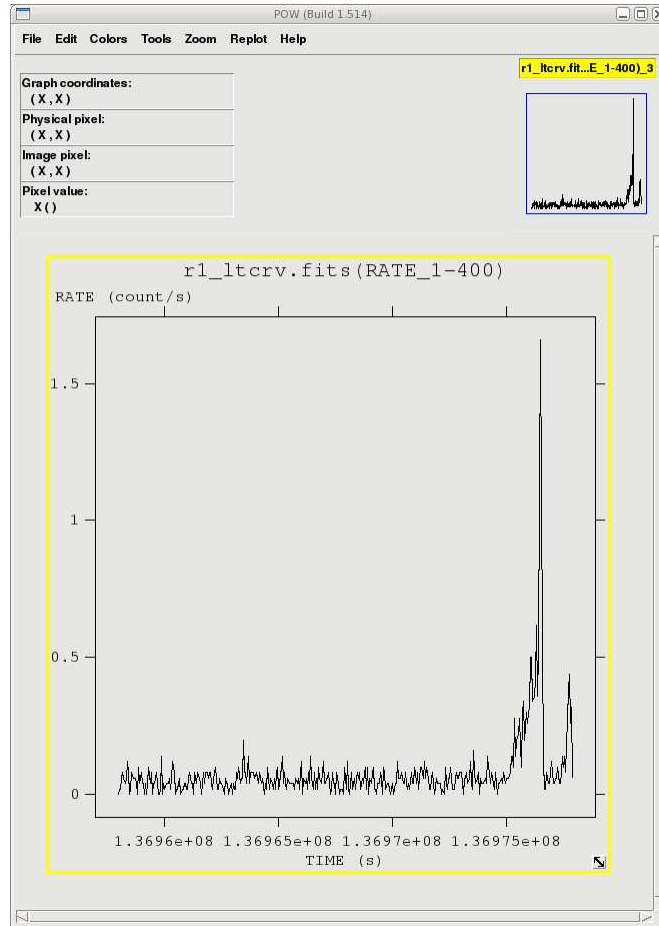
More experienced users should be aware that with SAS 13, the `*SRCLI*` file’s **column information changed**. *rgsproc* now outputs an `M_LAMBDA` column instead of `BETA_CORR`, and `M_LAMBDA` should be used to generate the light curve. (The `*SRCLI*` file that came with the PPS products still contains a `BETA_CORR` column if you prefer to use that instead.)

To create light curves,

- 1) Enter the filtering criteria in the “Selection expression” box at the top of the *xmmselect* GUI:
(CCDNR==9)&&(REGION(P0153950701R1S001SRCLI_0000_PPS.FIT:RGS1_BACKGROUND,M_LAMBDA,XDSP_CORR))
- 2) Check the round box to the left of the **time** entry.
- 3) Click on the “OGIP Rate Curve” button near the bottom of the page. This brings up the *evselect* GUI.
- 4) Click on the “Lightcurve” tab and confirm that the **withrateset** box is checked.
- 5) Change the **timebinsize** to a reasonable amount, e.g. 10 or 100 s, and change the default output file name in the **rateset** box to something appropriate, in this case, **r1_ltcrv.fits**.
- 6) Click on the “Run” button at the lower left corner of the *evselect* GUI.

The resultant light curve is displayed automatically using Grace and is shown in Figure 7.5.

Figure 7.5: Background event rate from the RGS1 CCD9 chip. The flares are solar events. The time units are elapsed mission time.



7.4.4 Generating the Good Time Interval (GTI) File

Examination of the lightcurve shows that there is a noisy section at the end of the observation, after 1.36975e8 seconds, where the count rate is well above the normal background count rate of ~ 0.05 count/second. There are two procedures that make the GTI file (*gtibuild* and *tabgtigen*) that, when applied to the event file in another run of *rgsproc*, will excise these sections.

The first method, using *gtibuild*, requires a text file as input. In the first two columns, refer to the start and end times (in seconds) that you are interested in, and in the third column, indicate with either a + or -

sign whether that region should be kept or removed. Each good (or bad) time interval should get its own line. In the example case, we would write in our ASCII file (named `gti.txt`):

```
1.36958e8 1.36975e8 +
```

and proceed to the SAS task *gtibuild*:

- 1) Call the *gtibuild* task in the SAS GUI.
- 2) Enter the name of the text file in the **file** box.
- 3) Enter the name of the output fits file in the **table** box.
- 4) Click “Run”.

Alternatively, we can make the GTI file with *tabgtigen* and filter on RATE (though we could just as easily filter on TIME):

- 1) Call the *tabgtigen* task.
- 2) Enter the name of the lightcurve file in the **table** box, in this case, `r1_ltcrv.fits`.
- 3) Enter the name of the output file in the **gtiset** box, in this case, `gti.fits`.
- 4) Enter the filtering expression in the **expression** box. Since the nominal count rate is about 0.05 count/sec, we will set the upper limit to 0.2 count/sec: `RATE<0.2`
- 5) Click “Run”.

7.4.5 Applying the GTI

Now that we have GTI file, we can apply it to the event file by running *rgsproc* again. *rgsproc* is a complex task, running several steps, with five different entry and exit points. It is not necessary to rerun all the steps in the procedure, only the ones involving filtering.

To apply the GTI to the event file,

- 1) Call the *rgsproc* in the SAS GUI.
- 2) In the “global” tab, make sure that the **orders** box is set for both orders, 1 2 . Use the pulldown menus for **entrystage** and **exitstage** to select 3:filter and 5:fluxing, respectively.
- 3) In the “filter” tab, enter the name of the GTI file, in this case, `gti.fits`, in the **auxgtitables** box.
- 4) In the “spectra” tab, click on the “rgsspectrum” tab, and make sure that **bkgcorrect** is set to **no**.
- 5) In the “angles” tab, make sure the **withmlambda** column is set to **yes**.
- 6) Click “Run”.

7.4.6 Creating the Response Matrices (RMFs)

Response matrices (RMFs) are not provided as part of the pipeline product package, so you must create your own before analyzing data. The task *rgsproc* generates a response matrix automatically, but as noted in §7.3.4, the source coordinates are under the observer’s control. The source coordinates have a profound influence on the accuracy of the wavelength scale as recorded in the RMF that is produced automatically by *rgsproc*, and each RGS instrument and each order will have its own RMF.

Making the RMF is easily done with the package *rgsrmfgen*. Please note that, unlike with EPIC data, it is not necessary to make ancillary response files (ARFS).

To make the RMFs,

- 1) Call the *rgsrmfgen* task in the GUI.

- 2) In the `spectrumset` box, enter the name of the spectrum file; it has the form `*SRSPEC*`.
- 3) In the `evlist` box, enter the name of the event list, `r1_evt1.fits`.
- 4) Set `emin` to 0.4, `emax` to 2.5, and `rows` to 5000.
- 5) Set `rmfset` to the output file name, in this case, `r1_o1_rmf.fits`.
- 6) Click “Run”.

At this point, the spectra can be analyzed. If you wish, skip the discussion on combining spectra (§7.4.7) and go straight to fitting the spectrum (§7.6.)

7.4.7 Combining Spectra

Spectra from the same order in RGS1 and RGS2 can be safely combined to create a spectrum with higher signal-to-noise if they were reprocessed using *rgsproc* with `spectrumbinning=lambd`, as we did in §7.3 (this also happens to be the default). The task *rgscombine* also merges response files and background spectra. When merging response files, be sure that they have the same number of bins.

To merge RGS1 and RGS2 spectra,

- 1) Call the *rgscombine* task.
- 2) In the `pha` box, enter the spectrum files to combine, separated by a space:
`P0153950701R1S001SRSPEC1001.FIT P0153950701R2S002SRSPEC1001.FIT`.
- 3) In the `bkg` box, enter the corresponding background files,
`P0153950701R1S001BGSPEC1001.FIT P0153950701R2S002BGSPEC1001.FIT`.
- 4) In the `rmf` box, enter the response files, `r1_o1_rmf.fits r2_o1_rmf.fits`.
- 5) In the `filepha` box, enter the output merged spectrum, `r12_o1_srspec.fits`. In the `filermf` box, enter the output merged response files, `r12_o1_rmf.fits`. In the `filebkg` box, enter the output merged background spectrum, `r12_o1_bgspec.fits`.
- 6) Set the `rmfgrid` parameter to the same as that used to make the RMFs, 5000.
- 7) Click “Run”.

The spectra are ready for analysis, so we can now prepare the spectrum for fitting.

7.5 Approaches to Spectral Fitting and the Cash Statistic

For data sets of high signal-to-noise and low background, where counting statistics are within the Gaussian regime, the data products above are suitable for analysis using the default fitting scheme in XSPEC, χ^2 -minimization. However, for low count rates, in the Poisson regime, χ^2 -minimization is no longer suitable. With low count rates in individual channels, the error per channel can dominate over the count rate. Since channels are weighted by the inverse-square of the errors during χ^2 model fitting, channels with the lowest count rates are given overly-large weights in the Poisson regime. Spectral continua are consequently often fit incorrectly, with the model lying underneath the true continuum level. This will be a common problem with most RGS sources. Even if count rates are large, much of the flux from these sources can be contained within emission lines, rather than the continuum. Consequently, even obtaining correct equivalent widths for such sources is non-trivial.

The traditional way to increase the signal-to-noise of a data set is to rebin or group the channels, since, if channels are grouped in sufficiently large numbers, the combined signal-to-noise of the groups will jump into the Gaussian regime. However, this results in the loss of information. For example, sharp features like an absorption edge or emission line can be completely washed out. Further, in the Poisson regime, the background spectrum cannot simply be subtracted, as is commonly done in the Gaussian regime, since this could result in negative counts. Therefore, rebinning should be reserved for fast, preliminary analysis of spectra without sharp

features, or for making plots for publication. When working on the final analysis for a low-count data set, the (unbinned) background and source spectra should be fitted simultaneously using the Cash statistic. (If fitting with XSPEC, be sure you are running v11.1.0 or later. This is because RGS spectrum files have prompted a slight modification to the OGIP standard, since the RGS spatial extraction mask has a spatial-width which is a varying function of wavelength. Thus, it has become necessary to characterize the BACKSCL and AREASCL parameters as vectors (i.e., one number for each wavelength channel), rather than scalar keywords as they are for data from the EPIC cameras and past missions. These quantities map the size of the source extraction region to the size of the background extraction region and are essential for accurate fits. Only Xspec v11.1.0, or later versions, are capable of reading these vectors, so be certain that you have an up-to-date installation at your site.)

Finally, a caveat of using the Cash statistic in Xspec is that the scheme requires a “total” and “background” spectrum to be loaded into Xspec. This is in order to calculate parameter errors correctly. Consequently, be sure not to use the “net” spectra that were created as part of product packages by SAS v5.2 or earlier. To change schemes in Xspec before fitting the data, type:

```
XSPEC> statistic cstat
```

For our sample spectrum, we will rebin and fit it with χ^2 statistics.

7.5.1 Spectral Rebinning

There are two ways to rebin a spectrum: the FTOOL *grppha*, or the RGS pipeline. *grppha* can group channels using an algorithm which bins up consecutive channels until a count rate threshold is reached. This method conserves the resolution in emission lines above the threshold while improving statistics in the continuum.

To rebin the spectrum with *grppha*, type

```
grppha
```

and edit the parameters as needed:

```
> Please enter PHA filename[] P0153950701R1S001SRSPEC1001.FIT
> Please enter output filename[] P0153950701R1S001SRSPEC1001.bin30.FIT
> GRPPHA[] group min 30
> GRPPHA[] exit
```

The disadvantage of using *grppha* is that, although channel errors are propagated through the binning process correctly, the errors in the original spectrum product is not strictly accurate. The problem arises because there is no good way to treat the errors within channels containing no counts. To allow statistical fitting, these channels are arbitrarily given an error value of unity, which is subsequently propagated through the binning. Consequently, the errors are overestimated in the resulting spectra.

The other approach, which involves calling the RGS pipeline after it is complete, bins the data during spectral extraction. The following rebins the pipeline spectrum by a factor 3.

To rebin the spectrum with *rgsproc*, type

```
rgsproc orders='1 2' rebin=3 rmfbins=5000 entrystage=4:spectra
      finalstage=5:fluxing bkgcorrect=no
```

where

```
orders – dispersion orders to extract
rebin – wavelength rebinning factor
rmfbins – number of bins in the response file; this should be greater than 3000
entrystage – entry stage to the pipeline
finalstage – exit stage for the pipeline
```

One disadvantage of this approach is that you can only choose integer binning of the original channel size. To change the sampling of the events, the pipeline must be run from the second stage (“angles”) or earlier:

```
rgsproc orders='1 2' nbetabins=1133 rmfbins=5000 entrystage=2:angles
      finalstage=fluxing bkgcorrect=no
```

where the parameters are as defined previously, and

nbetabins – number of bins in the dispersion direction; the default is 3400

The disadvantage of using *rgsproc*, as opposed to *grppha*, is that the binning is linear across the dispersion direction. Velocity resolution is lost in the lines, so the accuracy of redshift determinations will be degraded, transition edges will be smoothed, and neighboring lines will become blended.

7.6 Fitting a Spectral Model

SAS does not include fitting software, so HEASoft packages will be used, and all fitting tasks will be called from the command line. Now that we have a response file, we can fit the spectrum using Xspec. We will use the unmerged files for this demonstration. (If you are using data from a low-count data set, make sure you are using Xspec v 11.1.0 or later and see §7.5.)

To fit the spectrum, type

```
xspec
```

Enter the data, background, and response file at the prompts, and edit the fitting parameters as needed. Please note that in this example, we are using the output from *rgsproc*, not the PPS data that came with the data set, so the names are slightly different. If we were using the PPS data, the input spectrum would be **SBSPEC**. Since we did not include background correction when we ran *rgsproc*, we can correct for it now.

```
XSPEC> data P0153950701R1S001SRSPEC1001.bin30.FIT ! input data
XSPEC> back P0153950701R1S001BGSPEC1001.bin30.FIT ! input background
XSPEC> resp r1_o1_rmf.fits ! input response file
XSPEC> ignore **0.4 ! set sensible limits
XSPEC> model wabs*pow ! set spectral model to absorbed powerlaw
1:wabs:nH> 0.01 ! enter reasonable initial values
2:powerlaw:PhoIndex>2.0
3:powerlaw:norm>1.0
XSPEC> renorm
XSPEC> fit
XSPEC> cpd /xw
XSPEC> setplot wave
XSPEC> setplot command window all
XSPEC> setplot command log x off
XSPEC> setplot command wind 1
XSPEC> setplot command r y 0 6e-17
XSPEC> setplot command wind 2
XSPEC> setplot command r y -10 10
XSPEC> plot data chi
XSPEC> exit
```

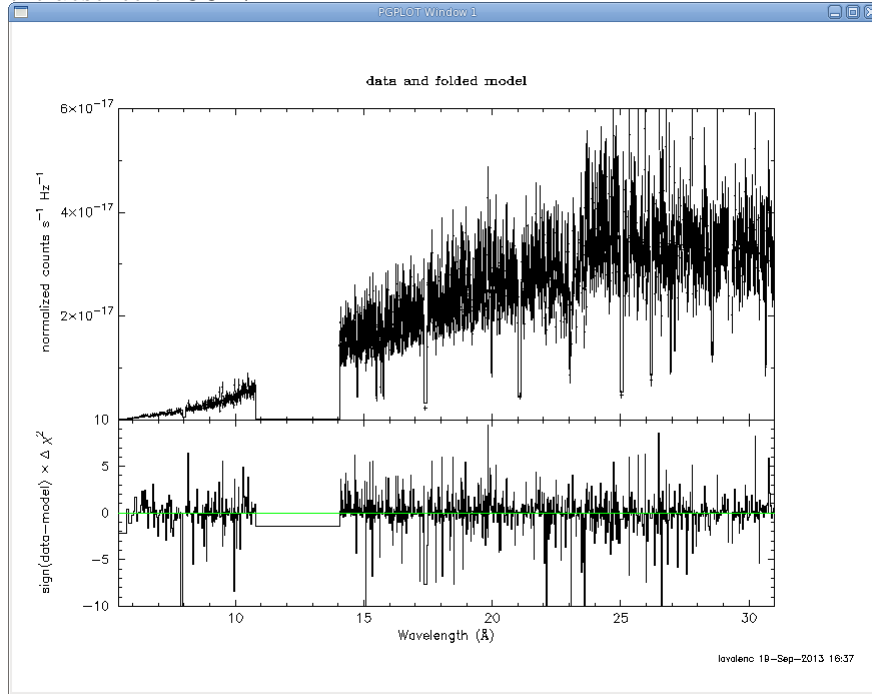
Figure 7.6 shows the fit to the spectrum.

7.7 Analysis of Extended Sources

7.7.1 Region masks

The optics of the RGS allow spectroscopy of reasonably extended sources, up to a few arc minutes. The width of the spatial extraction mask is defined by the fraction of total events one wishes to extract. With the default pipeline parameter values, 90% of events are extracted, assuming a point-like source.

Figure 7.6: 1st order RGS1 spectrum of Mkn 421. The fit is an absorbed power law model. The gap between 10–15 Å is due to the absence of CCD7.



Altering and optimizing the mask width for a spatially-extended source may take some trial and error, and, depending on the temperature distribution of the source, may depend on which lines one is currently interested in. While Mkn 421 is not an extended source, the following example increases the width of the extraction mask and ensures that the size of the background mask is reduced so that the two do not overlap.

Observing extended sources effectively broadens the psf of the spectrum in the dispersion direction. Therefore, it is prudent to also increase the width of the PI masks using the `pdistincl` parameter in order to prevent event losses.

To adjust the region mask with `rgsproc` from the command line, type

```
rgsproc orders='1 2' entrystage=4:spectra finalstage=5:fluxing bkgcorrect=no
      xpsfincl=99 xpsfexcl=99 pdistincl=95
```

where

`xpsfincl` – include this fraction of point-source events inside the spatial source extraction mask
`xpsfexcl` – exclude this fraction of point-source events from the spatial background extraction mask
`pdistincl` – include this fraction of point-source events inside the pulse height extraction mask

Alternatively, to do the same in the SAS GUI,

- 1) Call `rgsproc`.
- 2) In the “global” tab, make sure that the `orders` box is set for both orders, 1 2.
- 3) In the “global” tab, use the pulldown menus for `entrystage` and `exitstage` to select 4:spectra and 5:fluxing, respectively.
- 4) In the “spectra” tab, in the “rgsregions” sub-tab, set both `xpsfincl` and `xpsfexcl` to 99, and `pdistincl` to 95.
- 5) Click “Run”.

7.7.2 Fitting spectral models to extended sources

RGS response matrices are consistent for point sources only. Since extended source spectra are broadened, the simplest way to deal with this problem during spectral fitting is to reproduce the broadening function, and convolve it across the spectral model. Xspec v11.2 and later contains the convolution model `rgsxsrc`. It requires two external files to perform the operation:

- 1) An OGIP FITS image of the source. The better the resolution of the image, the more accurate the convolution. For example, if a Chandra image of the source is available, this will provide a more accurate result than an EPIC image.
- 2) An ASCII file containing three lines of input. For this example case, we will name it `xsource.mod`. It defines three environment variables and should look like this example:

```
RGS_XSOURCE_IMAGE ./MOS1.fit
RGS_XSOURCE_BORESIGHT 23:25:19.8 -12:07:25 247.302646
RGS_XSOURCE_EXTRACTION 2.5
```

where

```
RGS_XSOURCE_IMAGE – path to the source image
RGS_XSOURCE_BORESIGHT – RA, Dec of the center of the source and PA of the telescope
RGS_XSOURCE_EXTRACTION – The extent (in arcmin), centered on the source, over which you want to
construct the convolution function. You want this “aperture” to be larger than the source itself.
```

To set these environment variables within Xspec execute the command:

```
xset rgs_xsource_file xsource.mod
```

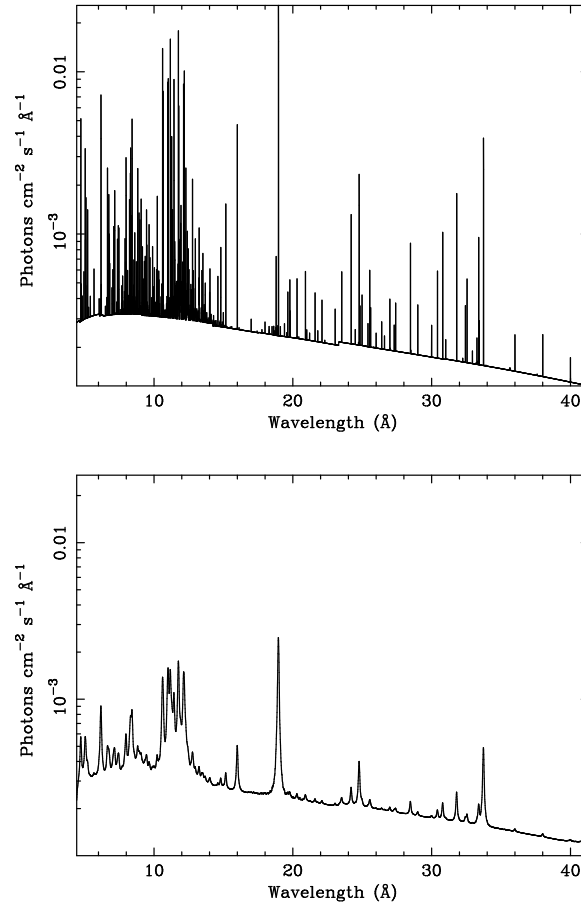
Here is an example. Note that the spectral order is always negative.

xspec

```
XSPEC>data P0108460201R1S004SRSPEC1003.FIT
XSPEC>ignore bad
XSPEC>xset rgs_xsource_file xsource.mod
XSPEC>model rgxssrc*wabs*mekal
rgxssrc:order>-1
wabs:nH>1
mekal:kT>2
mekal:nH>1
mekal:Abundanc>1
mekal:Redshift>
mekal:Switch>0
mekal:norm>1
XSPEC>renorm
XSPEC>fit
XSPEC>setplot device /xs
XSPEC>setplot wave
XSPEC>setplot command window all
XSPEC>setplot command log x off
XSPEC>plot data residuals
XSPEC>exit
Do you really want to exit? (y)y
```

Figure 7.7 compares a point source model with an extended source counterpart.

Figure 7.7: The top figure is a thin, thermal plasma at 2 keV from a point source. The lower figure is the same spectral model, but convolved by the MOS1 0.3–2.0 keV spatial profile of a low-redshift cluster.



7.7.3 Model limitations

Users should be aware that this method assumes an isothermal source (or uniform emissivity from line to line in the case of a non-thermal spectrum) where the spatial distributions of all the lines are identical. In reality, however, the thermal structure of the source is likely to be more complicated. The broad-band convolution function may bear little resemblance to the correct function for particular line transitions.

One way around this problem would be to have a temperature map of the source to define line emissivity across the source and convolve the model spectrum accordingly. The RGS instrument team at the Columbia Astrophysics Laboratory are developing a Monte Carlo code to perform an operation with this effect. While it is unlikely the code will be publicly available in the near future, the team welcomes investigators who would be interested in collaboration. Interested parties are encouraged to contact John Peterson (jrpeters@astro.columbia.edu).

Chapter 8

An OM Data Processing and Analysis Primer

As with EPIC and RGS datasets, many files are associated with an OM dataset. The `INDEX.HTM` file, and links therein, are viewable with a web browser and will help you navigate the dataset. The different types of files are discussed in Chapter 3.2; however, since the OM is somewhat different from the other instruments on-board XMM-Newton, we will discuss them in more detail in later sections.

The OM can operate in Imaging, Fast, and Grism mode. Each of these modes has dedicated commands to reprocess the data: *omichain*, *omfchain*, and *omgchain*. These are metatasks that each call several procedures that are used to prepare the data for processing, make and apply flatfield images, and detect sources. The tasks *omichain* and *omfchain* also calculate the instrumental magnitudes of sources, find the position of the sources (in equatorial coordinates), and produce a sky image; *omgchain* produces a spectrum. If you run these chains, it is helpful to inspect the `sas_log` file to get a detailed list of the performed tasks. These chains rely on filters specified by the user; if no arguments are given, they run on all the files present in the ODF directory. **Due to the long file names and the large number of input parameters, users are urged to simply use the chains and not run the chains' individual tasks one at a time.**

Most OM data are obtained in Imaging mode. If they were obtained in the Fast mode, there will be an additional event list file corresponding to the Fast window (`*FAE.FIT`). Reprocessing of data taken in Fast mode using the command line and SAS GUI is discussed in §8.3. Reprocessing OM Grism data is discussed in §8.4.

As always, it is **strongly** recommended that you keep all reprocessed data in its own directory! SAS places output files in whichever directory it is in when a task is called. Throughout this Guide, it is assumed that the Pipeline Processed data are in the PPS directory, the ODF data (with upper case file names, and uncompressed) are in the directory ODF, and the analysis is taking place in the PROC directory. It is also assumed that the data were prepared for processing (see §5). For example data, we will use observations of the Lockman Hole (Obs ID 0123700101; the same as for the EPIC walk-through) for Image mode, Mkn 421 (Obs ID 0411081601; NOT the same as for the RGS walk-through!) for Fast mode, and BPM 16274 (Obs ID 0125320801) for Grism mode, though any dataset with the appropriate mode will suffice.

8.1 OM Artifacts and General Information

Before proceeding with the pipeline, it is appropriate to discuss the artifacts that often affect OM images. These can affect the accuracy of a measurement by, for example, increasing the background level. Some of these can be seen in Fig. 8.1.

- Stray light – background celestial light is reflected by the OM detector housing onto the center on the OM field of view, producing a circular area of high background. This can also produce looping structures and long streaks.
- Modulo 8 noise – In the raw images, a modulo 8 pattern arises from imperfections in the event centroiding algorithm in the OM electronics. This is removed during image processing.
- Smoke rings – light from bright sources is reflected from the entrance window back on the detector, producing faint rings located radially away from the center of the field of view.

- Out-of-time events – sources with count rates of several tens of counts/sec show a strip of events along the readout direction, corresponding to photons that arrived while the detector was being read out.

Further, artifacts also can contaminate grism data. Due to this mode’s complexity, users are urged to be very careful when working with grism data, and should refer to the SOC’s website on this topic.

Users should also keep in mind some differences between OM data and X-ray data. Unlike EPIC and RGS, there are no good time intervals (GTIs) in OM data; an entire exposure is either kept or rejected. Also, OM exposures only provide direct energy information when in grism mode, and the flat field response of the detector is assumed to be unity.

For detailed descriptions of PP data nomenclature, file contents, and which tasks can be used to view them, see Tables 3.2 and 3.3.

If you simply want a quick look at your data, sky images and source lists are in `*SIMAGE*.FTZ` and `*SWSRLI*.FTZ`, respectively. Further, there are low resolution sky images for each filter; they follow the nomenclature:

PjjjjjjkkkkOMX000RSIMAGbb000.QQQ

jjjjjj – Proposal number

kkkk – Observation ID

b – Filter keyword: B, V, U, M (UVM2), L (UVW1) and S (UVW2)

QQQ – File type (e.g., PNG, FTZ)

So for example, P01237001010MX000RSIMAGV000.FTZ is the final low resolution sky image in the V filter. To see what files have been summed to make the final image, search for the keyword XPROCO in the FITS header. For our example image, this would be

```
XPROCO = 'ommosaic imagesets='product/P01237001010MS004SIMAGE1000.FIT produc&
CONTINUE 't/P01237001010MS415SIMAGE1000.FIT product/P01237001010MS416SIMAGE10&
CONTINUE '00.FIT product/P01237001010MS417SIMAGE1000.FIT product/P01237001010&
CONTINUE 'MS418SIMAGE1000.FIT' 'mosaicedset=product/P01237001010MX000RSIMAGV0&
CONTINUE '00.FIT exposuremap=no exposure=1000 # (ommosaic-1.11.7) [xmmsas_200&
CONTINUE '61026_1802-6.6.0]'
```

The source list file (`*SWSRLI*.FTZ`) also contains useful information for the user; the column names are listed in Table 8.1.

Table 8.1: Some of the important columns in the source list file.

Column name	Contents
SRCNUM	Source number
RA	RA of the detected source
DEC	Dec of the detected source
POSERR	Positional uncertainty
RATE	extracted count rate
RATE_ERR	error estimate on the count rate
SIGNIFICANCE	Significance of the detection (in σ)
MAG	Brightness of the source in magnitude
MAGERR	uncertainty on the magnitude

8.2 Imaging Mode

8.2.1 Rerunning the Pipeline

Please note that calling any of the repipelining tasks will initiate processing on all OM data of that particular mode; currently, only *omichain* will accept parameters to limit processing to a specific filter or exposure.

To rerun the pipeline on all exposures and filters from the command line, type

```
omichain
```

Alternatively, to run the pipeline from the SAS GUI,

- 1) Call *omichain*.
- 2) In the task pop-up window, click “Run”.

This produces numerous files, including images and regions for each exposure and each filter. If we are interested in the sources detected in the mosaicked V band image, we could run *omichain* with the appropriate flags from the command line typing:

```
omichain filters=V processmosaicedimages=yes omdetectnsigma=2.0
omdetectminsignificance=3.0
```

where

```
filters – list of filters to be processed
processmosaicedimages – process the mosaicked sky images?
omdetectnsigma – number of  $\sigma$  above background required for a pixel to be considered part of a source
omdetectminsignificance – minimum significance of a source to be included in the source list file
```

Alternatively, we can do it with the SAS GUI:

- 1) Call *omichain*.
- 2) In the pop-up window, in the “0” tab, next to “filters”, enter V; next to “omdetectnsigma”, enter 2.0; next to “omdetectminsignificance”, enter 3.0. In the “1” tab, next to “processmosaicedimages”, enter **yes**.
- 3) Click “Run”.

The output files can be used immediately for analysis, though users are strongly urged to examine the output for consistency first (see §8.2.2). The chains apply all necessary corrections, so no further processing or filtering needs to be done. Please note that the chains do not produce output files with exactly the same names as those in the PPS directory (they also produce some files which are not included in the PPS directory at all.) Table 8.2 lists the file ID equivalences between repipelined and PPS files.

8.2.2 Verifying the Output

While the output from the chains is ready for analysis, OM does have some peculiarities, as discussed in §8.1. While these usually have only an aesthetic effect, they can also affect source brightness measurements, since they increase the background. **In light of this, users are strongly encouraged to verify the consistency of the data prior to analysis.** There are a few ways to do this. Users can examine the combined source list with *fv*, which will let them see if interesting sources have been detected in all the filters where they are visible. Users can also overlay the image source list on to the sky image with *implot* by downloading the relevant files to the user’s local machine. This can also be done with *ds9* or *gaia* by using *slconv* to change source lists into region files and downloading the relevant files to your local machine. The task *slconv* allows users to set the regions radii in arcseconds to a constant value or scale them to header keywords, such as RATE. By default, *ds9* region files have suffixes of **.reg**; *gaia* region files have suffixes of **.gaia**. In the example below, we make a region file from the source list for the mosaicked, V-band sky image.

To make a *ds9* region file from a source list from the command line, type:

Table 8.2: File ID equivalences between repipelined and PPS OM files.

Repipelined Name	PPS Name	Description
EVLIST	none	Fast mode events list
FIMAG_	FIMAG_	combined full-frame image
FLAFLD	none	flatfield
FSIMAG	FSIMAG	combined full-frame sky image
HSIMAG	HSIMAG	full-frame HIRES sky image mosaic
IMAGE_	IMAGE_	image from any filter or Grism
IMAGE_	IMAGEF	Fast mode image
LSIMAG	LSIMAG	full-frame LORES sky image mosaic
OBSMLI	OBSMLI	combined observation source list
REGION	SWSREG	sources region file
REGION	SFSREG	Fast mode sources region file
REGION	SGSREG	Grism <i>ds9</i> regions
RIMAGE	GIMAGE	Grism rotated image
RSIMAG	RSIMAG	default mode sky mosaic
SIMAGE	SIMAGE	sky aligned image
SIMAGE	SIMAGEF	Fast mode sky aligned image
SIMAGE	none	Grism sky aligned image
SPCREG	SPCREG	Grism <i>ds9</i> spectrum regions
SPECLI	SPECLI	Grism spectra list
SPECTR	SPECTR	source extracted spectra
SUMMAR	SUMMAR	observation summary
SWSRLI	SWSRLI	sources list
SWSRLI	SFSRLI	Fast mode sources list
SWSRLI	SGSRLI	Grism sources list
TIMESR	TIMESR	Fast mode source timeseries
TSHPLT	TSHPLT	tracking history plot
TSTRTS	TSTRTS	tracking star timeseries

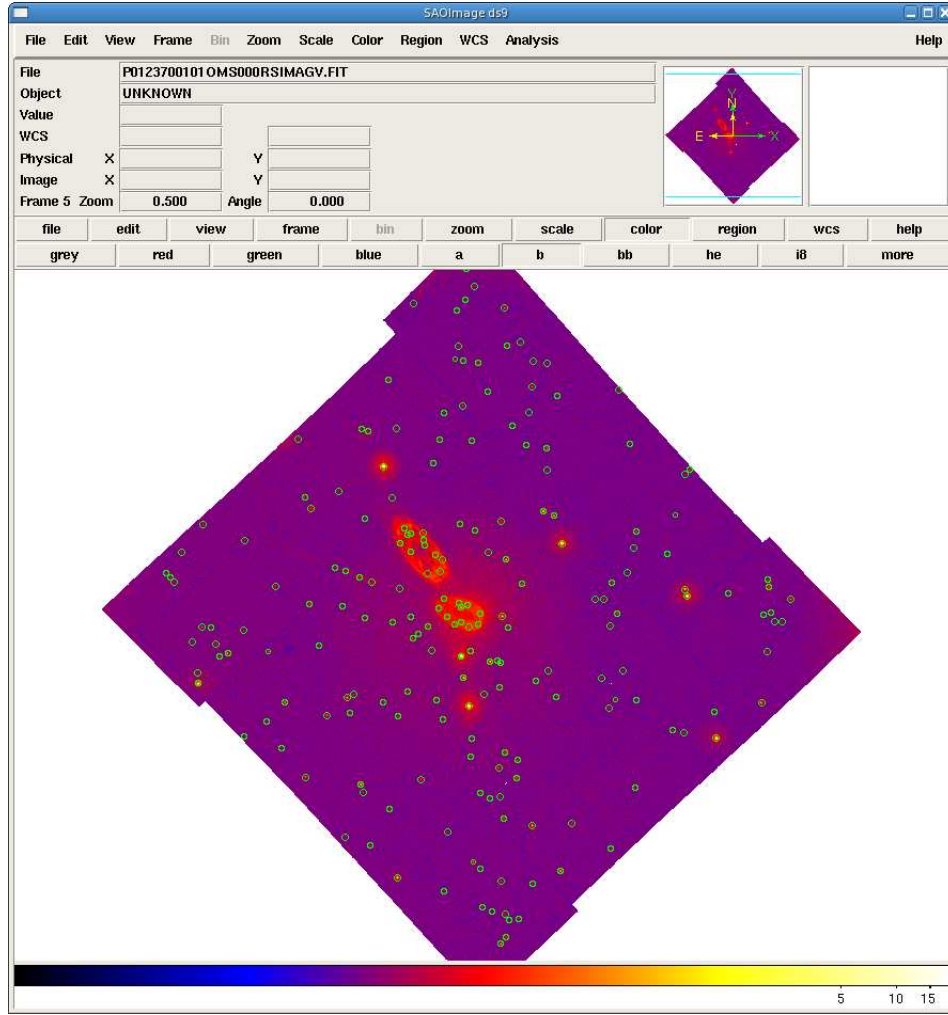
```
slconv srclisttab=P01237001010MS000RSISWSV.FIT radiusexpression=5
      outfileprefix=Vband_mosaic outputstyle=ds9
```

where

srclisttab – source list file name
radiusexpression – constant or expression (possibly involving keywords) used
to determine the radii of the plotted circles
outputstyle – output format; either *ds9* or *gaia*
outfileprefix – prefix of output file name

The mosaicked, V-band sky image, `P01237001010MS000RSIMAGV.FIT`, with the region file from *slconv* overlaid, is shown in Fig. 8.1. There clearly are spurious detections; these can be removed by hand, or by rerunning *omichain* with a different background-level threshold or source significance.

Figure 8.1: A mosaicked, V-band sky image and corresponding region file viewed with *ds9*. Spurious detections are clearly present and can be removed either by hand or by rerunning *omichain* with the appropriate flags. Some of the artifacts mentioned in §8.1 are also visible.



8.3 Fast Mode

8.3.1 Rerunning the Pipeline

The repipelining task for OM data taken in fast mode is *omfchain*. It produces images of the detected sources, extracts events related to the sources and the background, and extracts the corresponding light curves. At present, unlike *omichain*, *omfchain* does not allow for keywords to specify filters or exposures; calling this task will process all fast mode data.

To run the pipeline on fast mode data on the command line, type

```
omfchain
```

Alternatively, *omfchain* can be run from the GUI:

- 1) Call *omfchain*.
- 2) In the task pop-up window, click “Run”.

There are two types of output files: those that start with **f** are intermediate images or time series files; those that start with **p** are products. The output files are described in Table 8.2.

To demonstrate some of these output files, we have rerun the pipeline on the example dataset. The processed image in sky-coordinates from one exposure, `P0411081601OMS006SIMAGE1000.FIT`, is shown in Fig. 8.2 (left). The background-subtracted light curve produced automatically by the task, `F0411081601OMS006TIMESR1000.PS`, is shown in Fig. 8.2 (right).

8.3.2 Verifying the Output

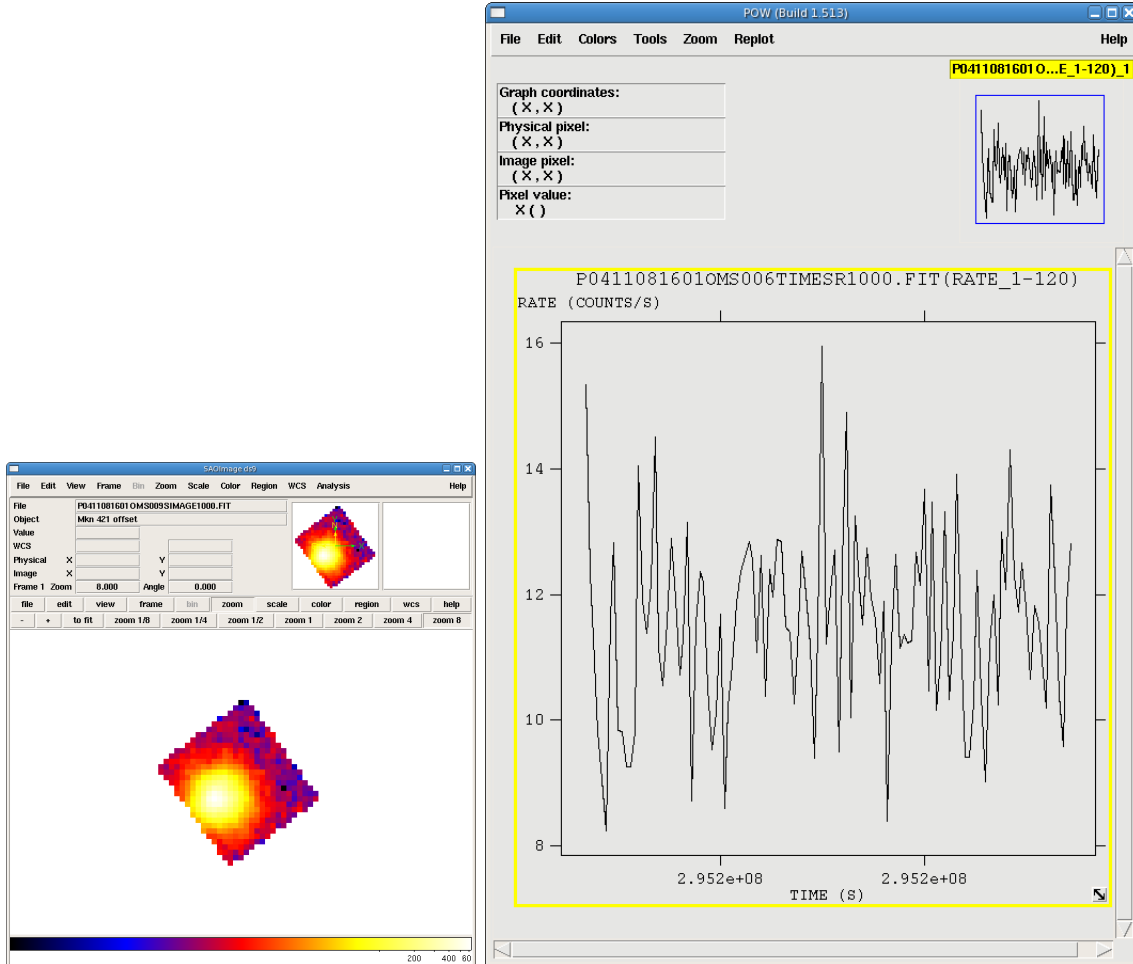
A good first step in checking the output is to examine the light curve plot for both the source and background, making sure they are reasonable: no isolated, unusually high (or low) values, and no frequent drop-outs. Users should also check the image with *fv* or *ds9* in the Fast mode window to see if the source is near an edge. If it is, it's a good idea to examine the light curves from different exposures to verify that they are consistent from exposure to exposure (while keeping in mind any intrinsic source variability). If the image is blurred or unusual in any way, users should check the tracking history file to verify the tracking was reliable.

8.4 Grism Analysis

8.4.1 Rerunning the Pipeline

The repipelining task for OM data taken in grism mode is *omgchain*. It produces images of the detected sources and background, extracts source spectra and region files, and makes source lists and postscript and PDF plots. At present, unlike *omichain*, *omgchain* does not allow for keywords to specify filters or exposures; calling this

Figure 8.2: Left: The processed Fast mode sky image. Right: the light curve produced automatically by *omfchain*.



task will process all grism mode data.

To run the pipeline on fast mode data on the command line, type

```
omgchain
```

Alternatively, *omgchain* can be run from the GUI:

- 1) Call *omgchain*.
- 2) In the task pop-up window, click “Run”.

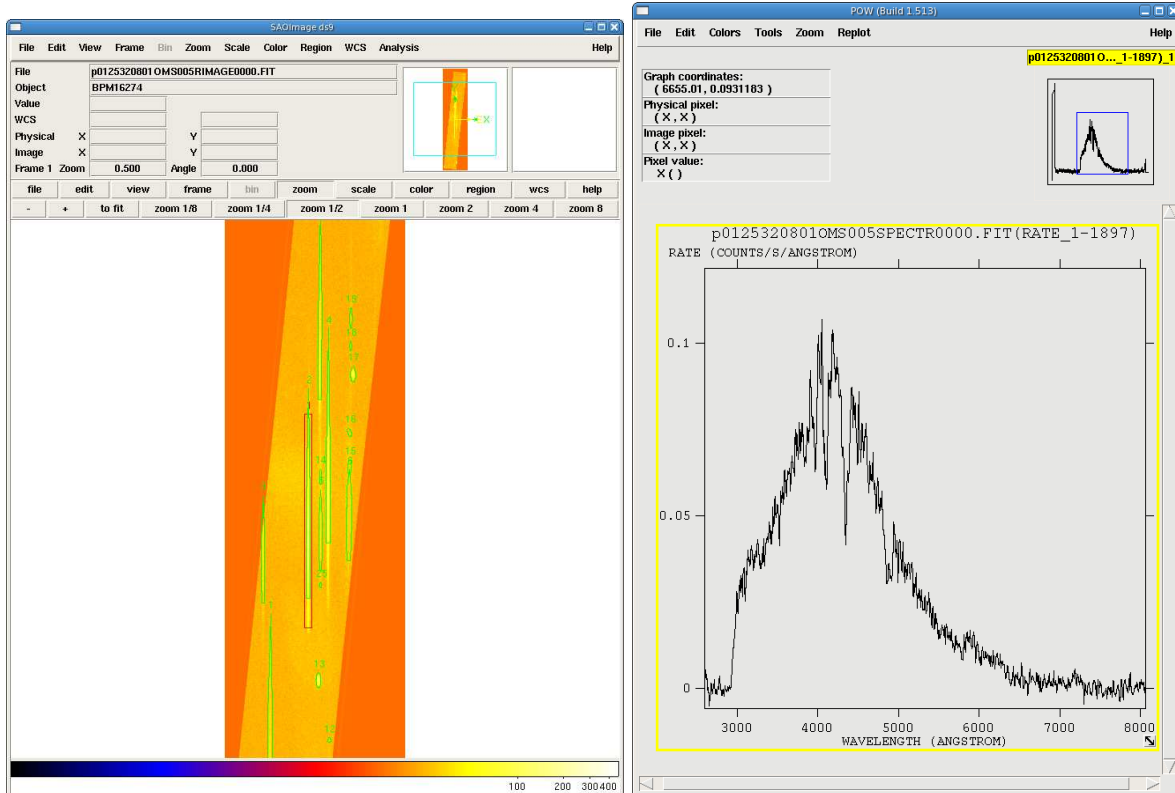
There are two types of output files: those that start with **g** are intermediate or auxiliary files and source lists; those that start with **p** are products. The output files are described in Table 8.2.

To demonstrate some of these output files, we have rerun the pipeline on the example dataset. The processed image, rotated to align with the columns of the image (p0125320801OMS005RIMAGE0000.FIT), is shown in Fig. 8.3 (left). Two region files are overlayed: p0125320801OMS005REGION0001.ASC, which corresponds to the sources detected in this rotated image (green), and p0125320801OMS005SPCREG0001.ASC, which corresponds to the sources in the spectra list file (red) and indicates the locations of the zero and first orders. The task *omgchain* automatically extracted the spectrum of the red region (p0125320801OMS005SPECTR0000.FIT); this is shown in Fig. 8.3 (right).

8.4.2 Verifying the Output

The correct correlation of zero and first orders is crucial for grism analysis. Users should inspect the rotated image with *fv* or *ds9* and verify the identification of the orders by overlaying the ***SPCREG*** region file, as shown in Fig. 8.3 (left); the ***SPECLI*** file also contains this information. If users are interested in all source

Figure 8.3: Left: The repipelined, rotated image with regions overlayed. Right: the spectrum extracted from the source (red region).



detections, the region file can also be overlayed and the full source list examined. Users should also examine the spectra plots automatically produced by *omgchain*, for both the source and background, making sure they are reasonable. For improved source detection, the parameter **nsigma** can be changed.